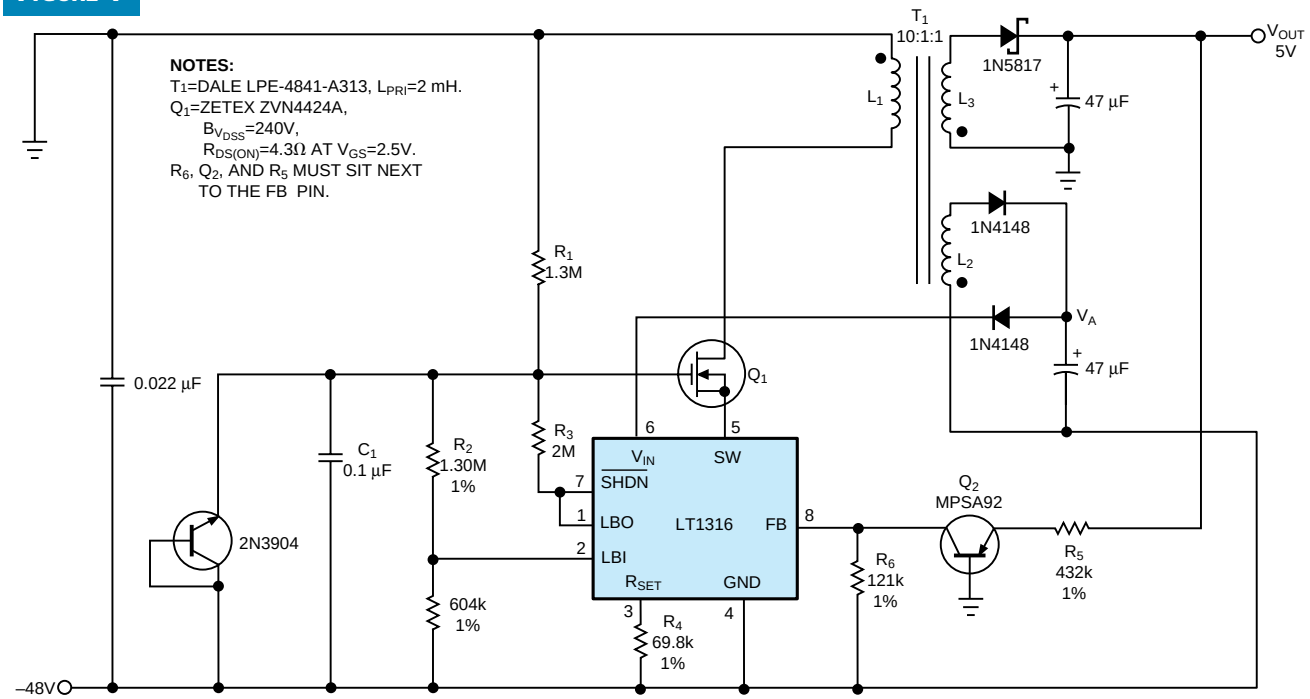
DC/DC converter operates from phone line

GARY SHOCKEY, LINEAR TECHNOLOGY CORP, MILPITAS, CA

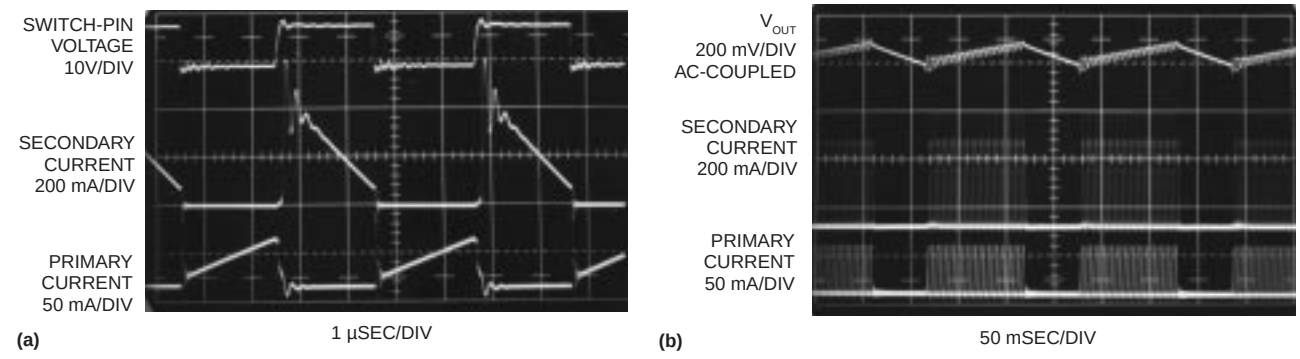
DC/DC converters for use inside the telephone handset require operation from the high-source-impedance phone line. Additionally, the CCITT specifications call for maximum on-hook power consumption of 25 mA. The dc/dc converter in **Figure 1** is 70%-efficient at an input power of 25 mA, providing 5V at 3.4 mA. Controlled, low-peak switch current ensures that the –48V input line experiences no excessive voltage drops during switching.

FIGURE 1



A –48-to-+5V flyback converter provides 3.4 mA of output current.

FIGURE 2



Switch voltage and current waveforms (a) show how the primary current ramps up during the switching cycle. The output ripple voltage is approximately 100 mV p-p (b).

The circuit operates as a flyback regulator with an auxiliary winding to provide power for the LT1316 switching-regulator IC. When you first apply power, the LBI pin is low, causing the SHDN pin to connect to ground through LBO. Grounding the SHDN pin places the part in shutdown mode, and only the low-battery comparator remains active. During this state, V_{IN} rises at a rate determined by R_1 and C_1. The IC draws only 6 μA in shutdown mode. R_1 needs to supply only this shutdown current, the current through R_2 and R_3, and C_1's charging current.

When LBI reaches 1.17V (which corresponds to a V_{IN} of approximately 3.7V), the LBO pin lets go of SHDN and the IC enters the active mode; switching action begins, and the output voltage begins to increase. As the device switches, the V_{IN} pin draws current out of C_1. V_{IN} then decreases sufficiently to trip the low-battery detector, stopping the switching. Start-up proceeds in this irregular fashion until, eventually, the voltage at V_A increases to 5V. (V_A is the same as V_{OUT} because L_2 and L_3 have the same number of turns.) After start-up, current flows to the IC from V_A rather than from the –48V rail, increasing efficiency. The circuit will not start if V_{OUT} is loaded before it reaches 5V.

During each switch cycle, current in the transformer primary ramps up until reaching the current limit (**Figure 2a**). The value of R_4 sets the peak switch current. The circuit uses a 69.8-kΩ resistor to provide a peak switch current of 50 mA; increasing R_4 decreases the current limit. Secondary peak current is approximately equal to the primary peak current multiplied by the transformer's turns ratio (**Figure 2b**). The FB pin has a sense voltage of 1.23V, and you can set V_{OUT} by the following formula:

$$V_{OUT} = 1.23\left(\frac{R_5}{R_6}\right) + 0.6V.$$

For the load currents of 4 to 80 mA, the circuit achieves a minimum of 70% efficiency. Less than 80 μA quiescent current flows when the converter supplies 0.5 mA over 36 to 72V. (DI #2148) EDN

To Vote For This Design, Circle No. 356

Transfer data frames over asynchronous RS-232C lines

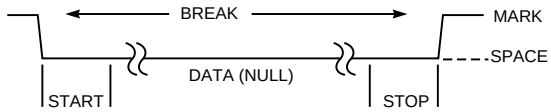
SK SHENOY, NAVAL PHYSICAL AND OCEANOGRAPHIC LAB, KOCHI, INDIA

The asynchronous RS-232C interface is a simple, low-cost option for interconnecting processor-based systems. In many applications, you need to transfer variable-size messages. However, the character-oriented RS-232C protocol offers no direct mechanism for transferring messages as self-contained packets. The method described here uses an obscure feature found in most UART devices to indicate packet boundaries. The feature is the capability to transmit and recognize the "Break" character. This character is nothing but a "space," or low, in the transmit line of a duration equal to or greater than an entire asynchronous character-transmission time, including the start and stop bits (**Figure 1**). In this framing method, the message data bytes sandwich between two Break characters to form a data frame (**Figure 2**).

A Turbo C program demonstrates the transfer of variable-size messages between two PCs with 8250-compatible UARTs (**Listing 1**). You can download the program from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the files from DI-SIG, #2140.

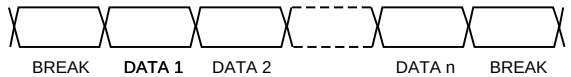
A null-modem cable interconnects the PCs' COM ports. The same routine works with most other UARTs. The method allows data-packet reception in interrupt mode and wastes no CPU overhead looking at each character to detect packet boundaries. Instead, the UART does the detecting. Because the Break is not a legitimate data character, it is data-

FIGURE 1



Most UARTs can transmit and recognize the Break character, a state of logic 0 between the start and stop bits.

FIGURE 2



The Turbo C routine in Listing 1 sandwiches data bytes between two Break characters to form a data frame.

transparent, and you can use it for binary-data exchange. You can use this "in-band" scheme with repeaters and modems, as long as they permit transmission of the Break condition. The packet-boundary detection is relatively immune to a missed Break character and to data errors. You can render the detection more robust by introducing data-length and check-sum fields in the frame to allow detection of errors and flow control using an RTS/CTS (request-to-send/clear-to-send) handshake.

To transmit a Break, set bit 6 (Set Break) of the line-control register to 1. The UART then sets its Tx line low, until bit 6 encounters a 0. Transmission of a Null character (00 hex) makes the duration of the Break equal to one character-transmission delay. Bit 6 of the line-status register (Tx Machine Status) indicates when this delay is over; then, the Break bit resets. To enable detection of the Break, bit 2 of the interrupt-enable register (interrupt-on-Rx-error condition) sets during UART initialization. Bit 0, set to 1, enables receive-data interrupts. In the interrupt-service routine (ISR), bits 1 and 2 of the interrupt-identification register indicate the interrupt type.

A global variable, Receive_Count, initialized to zero, handles frame reception. Upon detection of a Break, the UART raises an interrupt. If Receive_Count is zero, the interrupt is a start-of-frame break and the UART ignores it. (You can use the interrupt to set a Packet_Receive_On flag.) On each subsequent receive interrupt, the ISR stores the data in the Receive buffer with Receive_Count as the index. If Receive_Count is nonzero when the Break interrupt is raised, the interrupt is an end-of-frame break. Then the routine calls the frame-processing function and resets Receive_Count to zero. (DI #2140) EDN

To Vote For This Design, Circle No. 357

LISTING 1—DATA-FRAME-TRANSFER PROGRAM

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>

/* COM PORT DEFINITIONS AND GLOBAL VARIABLES */
#define   com_reg   0x3f8 /* Default is com1; 2f8 for com2 */
#define   DATA_PORT  com_reg + 0
#define   LINE_CNTRL com_reg + 3
#define   MODEM_CNTRL com_reg + 4
#define   INT_ENABL  com_reg + 1
#define   INT_IDENT  com_reg + 2
#define   LINE_STS  com_reg + 5
#define   MODEM_STS  com_reg + 6
#define   BAUD_LOW  com_reg + 0
#define   BAUD_HIGH  com_reg + 1
#define   DLAB_SET  0x80
#define   BAUDMSB   0
#define   BAUDLSB   0xc /* 9600 BPS */
#define   CNTRL_CMD  7  /* 8 BIT, 2 STOP BIT, NO PARITY */
#define   WAIT_TX_RDY() while (((inportb(LINE_STS))&0x60)!=0x60)
          /* Check for Tx buf empty & Tx shift reg empty */


unsigned char sdatabuf[256],rdatabuf[256]; /* Send & Recv buffers */
int Receive_Count = 0; /* Counter for data stored in rdatabuf[] */
void interrupt(*OldComHandler)(void);


/* FUNCTION CALLED TO DISPLAY RECEIVED DATA PACKET */
void processdata(void)
{
  int i;
  cprintf("\n\rRX > "); clreol(); /* Received data cursor */
  for (i = 0; i< Receive_Count; i++) /* Display received data */
              putch(rdatabuf[i]);
  cprintf("\n\rTX > "); clreol(); /* Transmitted data cursor */
 }


 /* INTERRUPT SERVICE ROUTINE TO TAKE CARE OF PACKET RECEPTION */
 void interrupt service_sio(void)
 {
  unsigned char iir ;
  iir = (inportb(INT_IDENT) >> 1) & 3; /* Get interrupt type */
  switch(iir)
  {
        case 0: /* Modem status int DSR,CTS,RI,RLSD */
          inportb(MODEM_STS); /* Ignore; reading IIR resets int */
          break;/* reading IIR resets int */
        case 1: /* Tx int */


          break;/* reading IIR resets int */
        case 2: /* Rx int */
          rdatabuf[Receive_Count++] = inportb(DATA_PORT); /* Store packet
          break;
        case 3: /* Rx error ( Break detect etc.) */
          inportb(DATA_PORT);   /* NULL char */
          if(((inportb(LINE_STS))&0x10) == 0x10)
          /* Break detected, Reading LSR Resets int */
            if (Receive_Count) processdata();/* EndOfFrame Break Process
            /* Else Start of Frame Break. Do nothing */
          /* Else Receive error; Drop packet */
          Receive_Count = 0; /* Re-initialise for next packet */
  }
  outportb(0x20,0x20); /* EOI */
  return;
}
```

```
/* FUNCTION TO INITIALISE SERIAL PORT */
void init_serial_io(void)
{
      outp(LINE_CNTRL,DLAB_SET);  /* DLAB_SET */
      outp(BAUD_LOW,BAUDLSB); outp(BAUD_HIGH,BAUDMSB); /* 9600 BAUD */
      outp(LINE_CNTRL,CNTRL_CMD); /* 8 BIT,2 STOP BIT,NO PARITY */
      outp(MODEM_CNTRL,8);     /* DTR,RTS & OUT2 SET */
      OldComHandler = getvect(0xc);/* 0xb for com2 */
      disable();
      setvect(0xc,(service_sio)); /* 0xb for com2 */
      outportb(0x21,((inportb(0x21))&(!0x10)));/* PIC mask word 0x8 for
      outportb(INT_ENABL,0x5); /* IER enable Rx Machine error & RX Data
      enable();
}


/* FUNCTION TO TRANSMIT A BREAK OF ONE CHARACTER DURATION */
void SendBreak(void)
{
   outportb(LINE_CNTRL,inportb(LINE_CNTRL) | 0x40); /* LCR; set break */
   outportb (DATA_PORT,0); /* Send NULL data */
   WAIT_TX_RDY(); /* Wait on TxShift Reg Empty; Null char is shifted out
   outportb(LINE_CNTRL,inportb(LINE_CNTRL) & 0xbf); /* LCR; remove break
}

/* FUNCTION TO TRANSMIT A DATA PACKET */
void SendBuffer(unsigned char packet[], int DatLen)
{
  int i;
  SendBreak(); /* Send START OF PACKET break */
  for (i=0; i<DatLen; i++) /* For each message byte*/
  {
    WAIT_TX_RDY(); /* Wait for Tx Ready */

    outportb (DATA_PORT,packet[i]); /* send one data char */
  }
  WAIT_TX_RDY(); /* Wait on TxShift Reg Empty; last char is shifted out
  SendBreak(); /* Send END OF PACKET break char */
}

/* BARE-BONES APPLICATION; TAKES STRING INPUT (TERMINATED BY ENTER) FROM
KEYBOARD AND TRANSMITS AS A PACKET. ALSO DISPLAYS RECEIVED PACKETS */
void main(void)
{
  int c, count = 0;

  init_serial_io(); /* Initialise serial port */
  printf("\n\rTX > "); /* Transmit Prompt */
  while(1) /* Forever Loop */
  {
      if((c=getche()) == 27) break; /* Exit if Escape key pressed */
      sdatabuf[count++] = c;
      if(c == '\r') /* If Enter Key pressed */
      {
        putch('\n'); clreol(); /* Go to newline */
        cprintf("TX > "); /* Transmit Prompt */
        SendBuffer(sdatabuf,count); /* Transmit Data Packet */
        count = 0; /* Reset Tx data count */
      }
  }
  setvect(0xc,(OldComHandler)); /* Restore int vector; 0xb for com2 */
  outportb(0x21,((inportb(0x21))|(0x10)));/* PIC mask word 0x8 for com2
}
```

Controller provides multiple alarm-driver formats

WILLIAM GRILL, RIVERHEAD SYSTEMS, LITTLETON, CO

Using a piezoelectric element for alarm applications offers low cost, low power, and flexibility. By coupling this element with a 12C508 programmable controller (Microchip Technology, Chandler, AZ), you can implement an eight-pin alarm generator. This approach provides multiple driver formats with a minimum of additional cost and footprint.

The controller in **Figure 1** drives the piezoelectric element directly from pins 2 and 7 with complementary square waves. A siren, chirp, warble, or constant-alarm output is available by setting the corresponding mode on pins 5 and 6 (**Table 1**). The design codes each mode as separate processes, which you can consider as variations of frequency, frequency step, and dwell.

The design also codes the positive and negative true alarm enables, pins 3 and 4, into the device. The controller retests the mode and these alarm enables at periodic intervals in the currently selected mode's cycle. This retesting permits dynamic selection of the output formats using the mode pins without a power reset. Applications can then use any or all of the output formats to indicate application alarm or status conditions.

The 12C508's internal RC oscillator provides the timing control used in each of the modes. Using an average of 3 mA, the controller operates from 2 to approximately 5.25V. The frequency-stepped formats are in constant "timebased" increments with constant frequency dwell times. Based on a 2.2-kHz piezo-element resonant peak, each of the mode's characteristics uses code-settable, dedicated registers to establish the output format.

The coded sequences use 127 bytes of code space. You can port the sequences into one of several code-compatible Microchip controllers or use a stand-alone peripheral controller, as in **Figure 1**, for any number of alarm applications. You can download applicable code from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2147. (DI #2147) EDN

To Vote For This Design, Circle No. 358

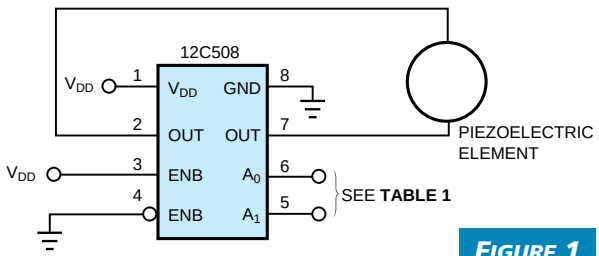


FIGURE 1

A programmable controller and a piezoelectric element combine to produce an alarm generator with multiple output formats: continuous frequency, two chirps, a warble, or a siren.

TABLE 1—STRAPPED ALARM CONFIGURATIONS

A_1	A_0	Alarm output
0	0	Continuous 2.2 kHz
0	1	Two single 2.2-kHz chirps: 0.2 sec each with 0.2 sec between
1	0	Warble: eight frequency steps swept continuously from below to above to below 2.2 kHz
1	1	Siren: repeating six frequency steps beginning at 2 kHz and stepping up to approximately 4.1 kHz

Algorithm extracts cube root

JOHN T HANNON JR, PHILIPS CONSUMER ELECTRONICS CORP, KNOXVILLE, TN

The C routine in **Listing 1** generates the cube root of either a positive or a negative number. The number can range from a small fraction to greater than 1 billion. Note that this idea is irrelevant for a PC, which includes a math library with the compiler and produces a more accurate result with less effort. However, this idea is useful if you use a processor for which you don't have a math library.

The main routine inputs a number from the keyboard and calls the *cuberoot* function. After calculating the cube root, the routine prints the result on the screen. The routine then recalculates the cube of this cube-root answer and prints this number on the screen so you can compare the accuracy of

the result with the original number. Five decimal places are set up for the print command to allow an accurate comparison.

This algorithm is similar to the square-root algorithm that uses the divide-and-average technique to reach a minimum error value. Before the cube-root operation begins, the routine determines if the number is positive or negative. If the value is negative, the program sets a flag so the returned root value can be set to a negative value. To start the process, the routine sets up an error value and assigns an initial value for the integer *root*. The error value for this function is 0.00001, and the initial value of *root* is 2.0. The routine squares this value and divides it into the given number. The routine then adds the result to the original value of *root* and divides this sum by 2 to generate the new value of *root*. This process continues until the absolute value of the difference between the cube of the value of *root* and the given number is less than the error value—in this case, 0.00001.

For some values, this algorithm approaches but never reaches the set error. Thus, the routine increments a counter after each iteration. If the counter reaches the maximum set count before the routine finds the minimum error, the function ends and returns the value of *root*. If the original number was negative, the routine changes the cube root to a negative value.

With the values in **Listing 1**, the accuracy for numbers from 0.1 to greater than 1 billion is better than 0.001%. For numbers less than 0.2, the accuracy is slightly lower. However, this accuracy difference is a result of the error value that you use in the calculation. By setting the error value (float variable *error*) to a smaller value, such as 0.000001, the accuracy for very small numbers can also approach this value.

Two other factors affect the accuracy: the number of bits the processor uses for floating-point numbers, and the size of the original number. For numbers with small absolute values, the variable *error* controls the accuracy. The accuracy, which is

(calculated root–actual root)/actual root,

is approximately equal to

error variable/3×original number.

So, with *error* set to 0.00001, the accuracy is about 0.003% for the cube root of 0.1, 0.03% for the cube root of 0.01, 0.3% for the cube root of 0.001, and so forth. In the limit—the cube root of 0—you can't divide by 0 to find percent accuracy, but the actual root that the program calculates is 0.0156250.

For numbers with large absolute values, the number of iterations determines the accuracy. For instance, the algorithm calculates the cube root of 10^{12} as 10083.87109 after 25 iterations and as10000.00000 after 45 iterations. (DI #2144) EDN

To Vote For This Design, Circle No. 359

LISTING 1—CUBE-ROOT EXTRACTOR

```
/* Function to calculate the cube root of a number. */

#include <stdio.h>

float absval(float x)            /* GENERATE THE ABSOLUTE VALUE */
{
  if (x < 0)
    x = -x;
  return (x);
}

float cuberoot (float num)                /* CUBE ROOT FUNCTION */
  {
  float error = 0.00001;                /* SET UP MAXIMUM ERROR */
  float root = 2.0;                       /* SET STARTING VALUE */
  int negflag = 0, count = 0;

  if (num < 0)                          /* IF NUMBER IS NEGATIVE */
    {
    num = -num;                          /* CHANGE TO POSITIVE & */
    negflag = 1;                         /* SET NEG. NUMBER FLAG */
    }

  while (absval(root * root * root - num) >= error)
    {
    root = (num/(root * root) + root)/2.0;
    ++count;
    if (count > 25)              /* IF NO MINIMUM ERROR AFTER 25 */
      break;                    /* ITERATIONS, EXIT THE FUNCTION */
    }            /* THIS COULD BE LARGER FOR VERY LARGE NUMBERS */

  if (negflag == 1)          /* IF ORIGINAL NUMBER WAS NEGATIVE */
    root = -root;               /* SET ROOT TO NEGATIVE NUMBER */
  return (root);
}

main()
{
  float number, root, newnum;

  clrscr();
  printf ("\n\n");
  printf ("\tEnter the number for cubed root:  ");
  scanf ("%f", &number);          /* INPUT A VALUE FOR CUBEROOT */

  root = cuberoot (number);          /* CALL CUBE-ROOT FUNCTION */
  printf ("\n\n\tThe cube root of %.3f is %.5f.",number, root);

  newnum = root * root * root;    /* CUBE ANSWER FOR COMPARISON */
  printf ("\n\n\n\t%.5f cubed is %.3f.",root, newnum);
  return(0);
}
```

Shunt regulator provides overvoltage protection

ROBERT N BUONO, MAHWAH, NJ

The circuit in **Figure 1** uses a typical technique for varying the output voltage of a power supply via a programmable control voltage. Although the topology and schematic details of the power supply are not critical, the protection technique is novel.

The control IC is a UC3843AN PWM controller. This IC applies 2.5V to the noninverting input of an internal error amplifier but does not bring this input out to a pin (node B). The inverting input of the error amplifier is available at an external pin (node A). To regulate V_{OUT}, the control IC must maintain the voltage at Node A equal to the 2.5V at Node B. The component values in the **figure** allow the dc output voltage of the power supply to vary between a minimum of 5V and a maximum of 75V, as a function of $V_{CONTROL}$, which can vary from 0 to 3V (corresponding to a V_{OUT} of 5V and 75V, respectively).

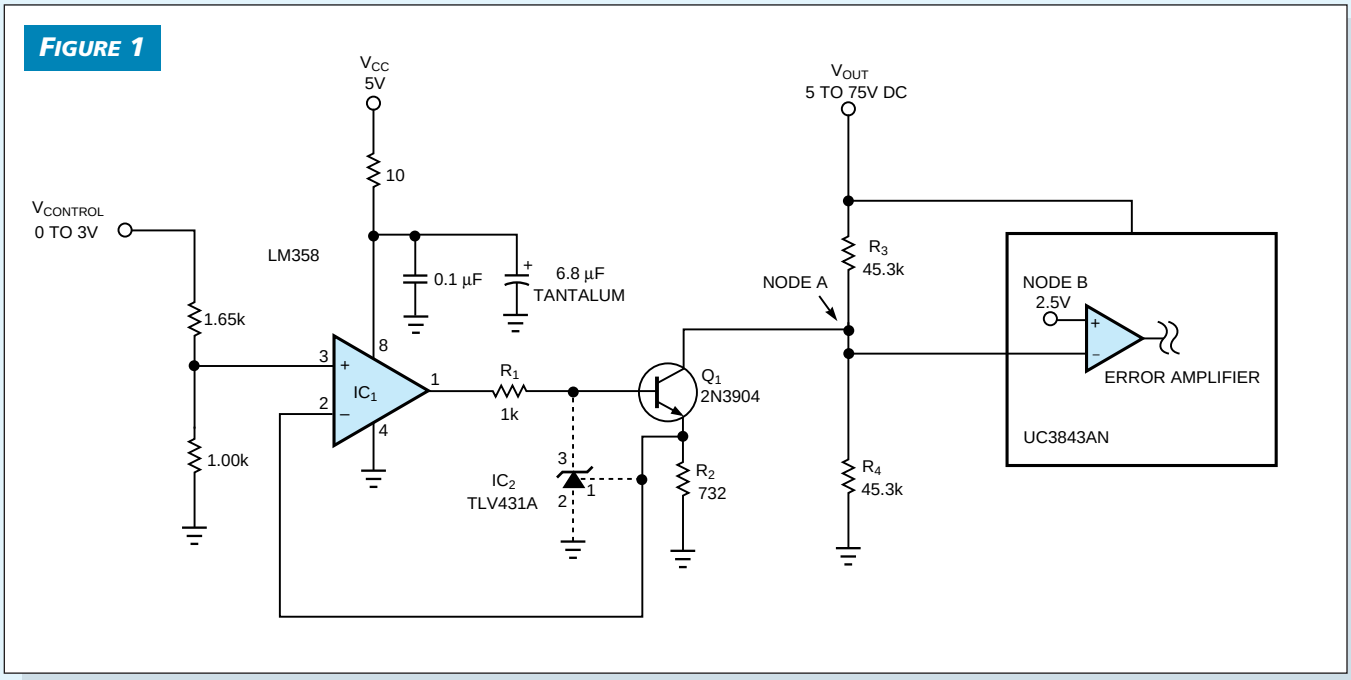
In the absence of Q_1, V_{OUT} and the voltage division of R_3 and R_4 determine the voltage at Node A. The circuit compares this voltage with the 2.5V at Node B. The power supply's output-voltage control loop keeps the voltage at Node A equal to Node B by appropriately controlling V_{OUT}.

The circuit provides programmability of V_{OUT} by sinking current from Node A. V_{OUT} must source any current that flows from this node. Also, the current must flow through R_3, which causes the voltage drop across R_3 to increase. V_{OUT} is then always equal to the voltage drop across R_3 plus a current set by $V_{CONTROL}$. IC_1's op amp forces the voltage across R_2 to equal the voltage at Pin 3 of IC_1.

The addition of just one component, IC_2, adds precise overvoltage protection to the variable-output power supply. IC_2 is a low-voltage shunt regulator that incorporates an internal 1.24V precision reference. This low reference voltage allows you to use this protection circuit with conventional power-supply control ICs for which 2.5V is a common internal reference voltage. Under normal operating conditions (for output voltages between 5 and 75V), IC_2 does nothing. The voltage at IC_2's reference (Pin 1) is less than its internal 1.24V reference, so its cathode (Pin 3) draws no current. In this case, IC_1 solely controls the voltage at the base of Q_1. For example, if $V_{CONTROL}$ is 3V, then the voltage across R_2 is 1.13V and V_{OUT} equals 75V. Note that, to simplify this example, the beta of Q_1 is assumed to be infinite.

However, in the event of any kind of fault that might cause the voltage across R_2 to rise above 1.24V (which corresponds to a maximum V_{OUT} of 81.7V), the shunt regulator begins to function. As the voltage at Pin 1 of IC_2 begins to exceed the internal 1.24 reference voltage, the cathode of IC_2 begins to conduct. The cathode of IC_2 then pulls down on the base of Q_1 to maintain 1.24V at Pin 1 of IC_2. As this happens, IC_2 through Q_1 controls the voltage across R_2. This con-

FIGURE 1



Adding one shunt regulator, IC_2, to an otherwise typical programmable power supply provides precise overvoltage protection.

trol causes the output of IC_1 to saturate at a positive voltage of approximately 3.7V because IC_1 can no longer keep the voltage across R_2 equal to the voltage at its input (Pin 3). The benefit to circuit operation is that IC_2 now operates with a constant-cathode current determined by the voltage across R_1, which equals 3.7–1.8V/1kΩ=1.9 mA. This level of cathode current ensures that IC_2 regulates properly.

This protection circuit is immune to any potential failure mode of IC_1's op amp or the programming voltage source, $V_{CONTROL}$. IC_1 operates only from 5V. If its output (Pin 1) shorts to ground, the minimum V_{OUT} results. If its output shorts to V_{CC}, IC_2 sinks 5–1.8V/1 kΩ=3.2 mA, and V_{OUT} clamps at the maximum of 81.7V. (DI #2146) EDN

To Vote For This Design, Circle No. 360

LED flasher and triac pulser work off ac line

DENNIS EICHENBERG, PARMA HEIGHTS, OH

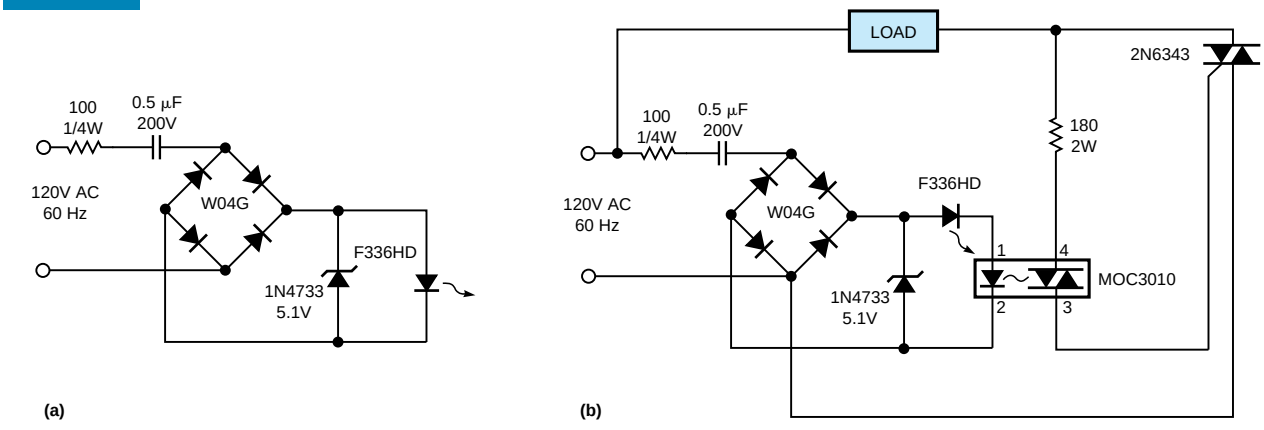
A flashing LED is an excellent visual alarm. Unfortunately, the LED is a dc device and requires additional circuitry to operate from an ac source. Several circuits can perform the necessary function, but the circuit in **Figure 1a** is the most efficient. This circuit is also reliable, compact, and inexpensive.

The F336HD red-flashing LED (part no. 276-036 at Radio Shack) operates directly from 5V and produces a consistent pulse of light at approximately 1 Hz without a time-constant capacitor. This LED starts immediately when you apply power and is insensitive to temperature variations. The W04G full-wave bridge rectifier produces a full-wave dc waveform from the 120V-ac line. The 0.5-µF capacitor provides current limiting for operating the LED from the rectified 120V-ac line. The 100Ω resistor protects the circuit from surges when you first apply power. The 1N4733 5.1V zener diode protects the LED from high-voltage excursions.

Some applications require a more intense alarm. A simple triac pulser can pulse a 120V-ac lamp or other resistive load of as much as 8A (**Figure 1b**). This circuit is also reliable, compact, efficient, and inexpensive. The circuit is similar to the one in **Figure 1a**, but, in this case, the F336HD LED drives an MOC3010 opto-coupled triac driver. The 180Ω resistor provides current limiting for the 2N6343 triac gate. This configuration can pulse a load as high as 960W. You can increase the power rating by choosing a different triac. (DI #2143) EDN

To Vote For This Design, Circle No. 361

FIGURE 1



A full-wave bridge rectifier provides a dc signal for the red-flashing LED (a). Adding a triac allows the circuit to pulse a 120V-ac lamp or other resistive load of as much as 8A (b).

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____ Fax _____

E-mail _____

My e-mail address may be published Yes____ No____

Company _____

Address _____

_____

_____

Country _____ ZIP _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

The winning Design Idea for the August 15, 1997, issue is entitled "Gates provide low-cost sine-wave generator," submitted by Adolfo Mondragon of Philips Components (Juarez, MX).

EDN | DESIGN IDEAS

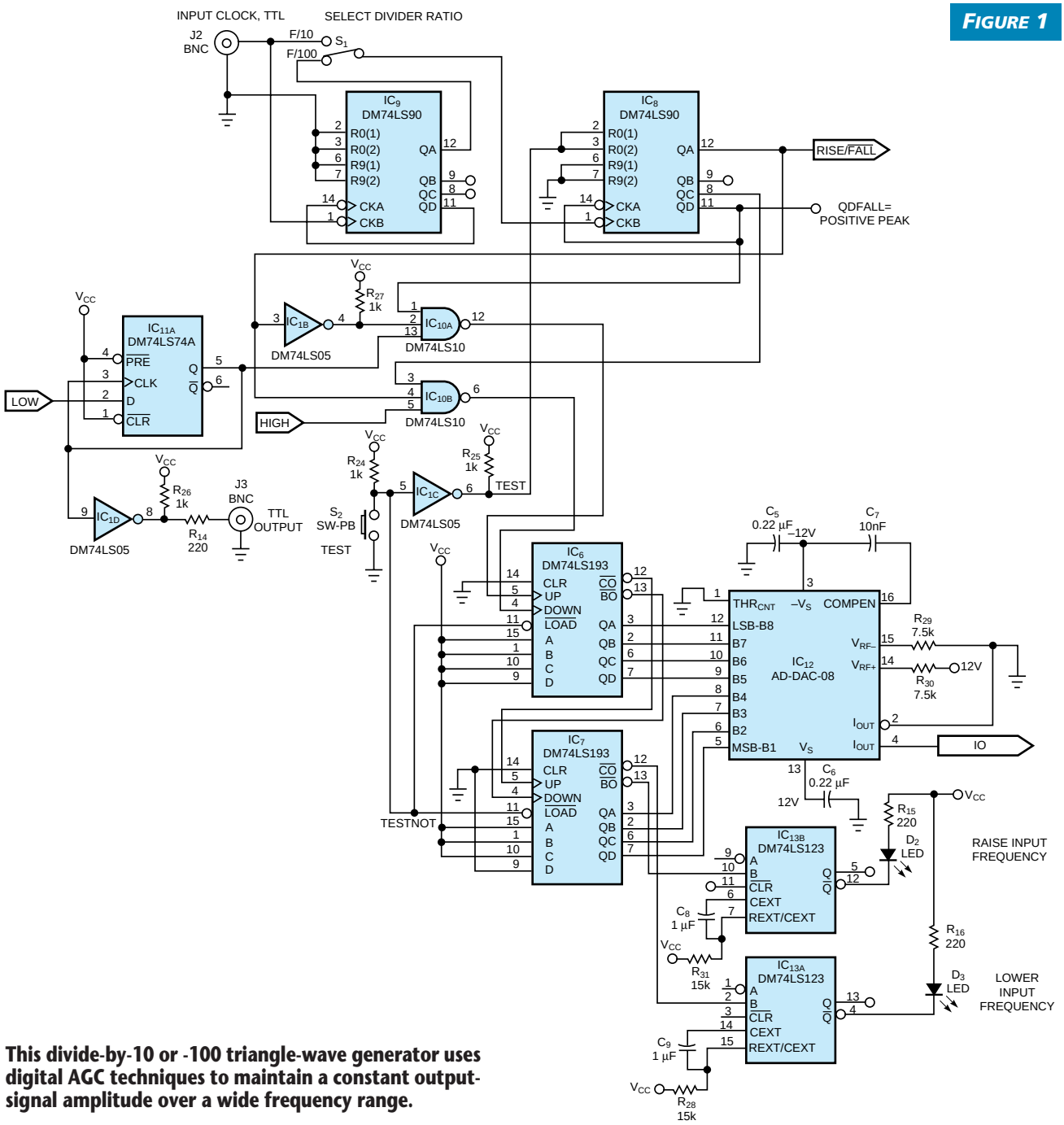
EDITED BY BILL TRAVIS & ANNE WATSON SWAGER

Scheme yields frequency-locked triangle waves

DANIEL DUFRESNE, DECATRON, ST BRUNO, PQ, CANADA

The circuit in **Figures 1** and **2** generates frequency-locked triangle waves of constant amplitude. It uses readily available TTL and other older-technology parts. The circuit portion in **Figure 1** comprises the frequency dividers, counters, and converter. The portion in **Figure 2** contains current sources, output circuits, and comparators. The circuit satisfies a need

FIGURE 1



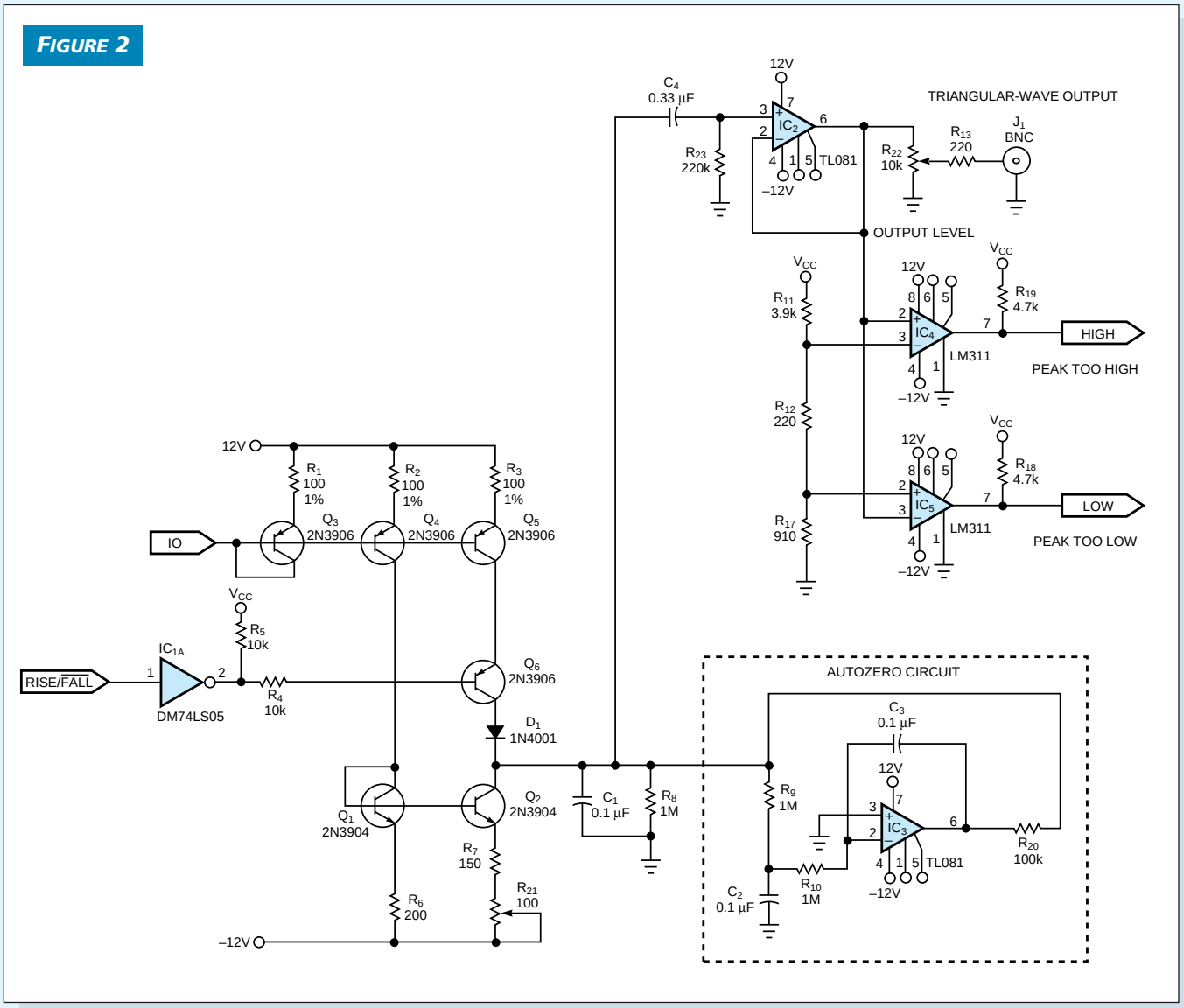
This divide-by-10 or -100 triangle-wave generator uses digital AGC techniques to maintain a constant output-signal amplitude over a wide frequency range.

for a triangle wave of approximately constant amplitude, at a frequency exactly one-tenth or $^1/_{100}$ that of an input TTL clock. The input frequency can vary over a wide range before the circuit loses lock. The triangle-wave output frequency ranges from 250 Hz to 2.5 kHz. Two decade dividers divide the input clock by 10 or 100. Switch S_1 selects the divider ratio. The signal RISE/$\overline{\text{FALL}}$ is a 50%-duty-cycle wave that turns on a 2I current source, comprising transistors Q_3, Q_4, and Q_5; a –I current source comprising Q_1 and Q_2 is always on.

After you initially adjust R_{21}, op amp IC_3 keeps any dc offset (caused by tracking imbalances between the current source and sink) to a minimal value. The output signal from the 8-bit DAC, IC_{12}, controls the current level in both the source and sink. The currents sum on capacitor C_2, thus generating a triangle wave. An op amp buffers the triangular signal; potentiometer R_{22} adjusts the output level. Assuming a fixed input frequency and a fixed capacitor value, you can adjust the current sources for the desired amplitude.

If the frequency decreases, the triangle-wave amplitude increases. To keep the amplitude constant, comparator IC_4 detects that the positive-peak amplitude is too high. The comparator sends input-clock signals to decrement the 8-bit counter IC_6 and IC_7, thus controlling the source- and sink-current value through the DAC. Similarly, if you raise the clock frequency, the triangle-wave amplitude decreases. Comparator IC_5 detects that the positive-peak amplitude at the end of the triangle wave's rise is too low. IC_{11a} latches the comparator output and gates clock pulses to increase the count and the source- and sink-current values.

FIGURE 2



A companion to Figure 1's converter, this block contains the supporting current sources, output circuits, and comparators.

If you run out of counts on IC_6 and IC_7, the carry and borrow output signals trigger the monostables, IC_{13A} and IC_{13B}. LED D_2 or D_3 lights to warn you to change the input frequency accordingly. Frequency tracking is asymmetric: As soon as the circuit detects the high limit, pulses decrement the counter. The sooner the high limit occurs, the more decrementing pulses the counter receives. However, you detect that the current is too low only when the RISE/$\overline{\text{FALL}}$ signal falls, and the triangular wave is still below the positive-peak low limit. This occurrence triggers a single pulse.

To adjust the circuit, set the divider to f/10, apply a 10-kHz TTL signal to the clock input, short-circuit C_2, push the TEST button, and adjust R_{22} for a zero-centered triangular wave. Release the TEST button, and remove the short circuit. You can increase the frequency range by using a bigger counter and a higher resolution DAC or by band-switching the current-summing capacitor, C_1, with some added logic that the borrow and carry outputs of IC_7 trigger. (DI #2127) EDN

To Vote For This Design, Circle No. 327

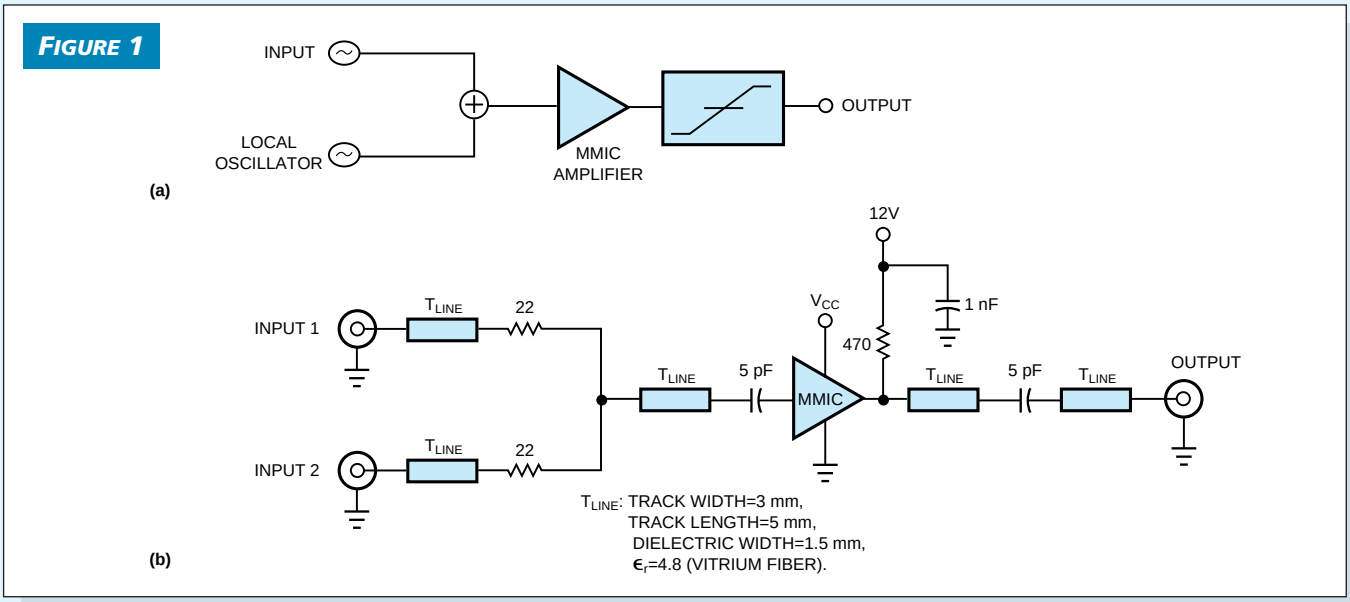
Off-the-shelf MMIC suits mixer applications

J LEIRA PAZ, M PEREIRA VARELA, AND FP FONTAN, UNIVERSITY OF VIGO, SPAIN

You can use a standard monolithic-microwave IC (MMIC) to configure an efficient and economical microwave mixer. Some systems require a mixer with an input level lower than –10 dBm. Many available mixers, however, require greater than 10-dBm input power. The circuit in **Figure 1** exploits the inherent nonlinearity of a Mini-Circuits (Brooklyn, NY) MAR6 microwave amplifier to configure a low-level mixer. The idea is to add the input signals and feed the sum to the MMIC (**Figure 1a**). The amplifier's saturation characteristic produces the harmonics and intermodulation products. The MMIC has greater than 10-dB gain at frequencies over 1 GHz, a desirable characteristic for making mixers for the 0.9, 1.8-, and 2.5-GHz bands.

The MMIC also has a maximum power output of 2 dBm at 1-dB compression, a useful trait for obtaining intermodulation products with low input power. The gain of the MMIC varies from 17 dB at 100 MHz to 10 dB at 2.4 GHz. The gain can thus generate intermodulation products using two input signals whose power is lower than –10 dBm. Finally, the MMIC's 3-dB noise figure is an important factor for low-level input signals. The adder uses two standard-value 22Ω resistors and thus presents a reasonably good impedance match in 50Ω systems (**Figure 1b**). For perfect impedance matching, you would need 20.7Ω (unavailable) resistors. With the 22Ω resistors, the input VSWR is less than 1.4 at 1 GHz. To obtain optimum results, you should use microstrip techniques, with the dimensions in **Figure 1b**, in designing the pc board.

FIGURE 1



A summing network and a standard MMIC (a), connected with microstrip techniques (b), produce an efficient mixer for low-level signals.

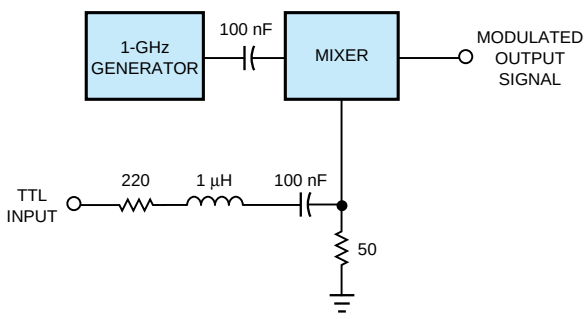
You can use this mixer in a variety of applications: downconverters and modulators, for example. Another example is an amplitude-shift-keying (ASK) modulator for digital applications (**Figure 2**). The mixer block in **Figure 2** uses the circuit in **Figure 1a**. The measurements of **Figures 3**, **4**, and **5** reflect a 1-GHz carrier frequency and a 4-Mbps digital signal. To obtain a good relationship between the carrier level and the modulated-signal level, use a –5-dBm input-carrier level, a ±0.5-dBm digital signal, and a 12V supply. The output-signal level is 0 dBm.

Figure 3 shows the magnitude of the intermodulation products as a function of the 100-MHz signal power. The input frequencies are 1.3 GHz and 100 MHz. The most important intermodulation frequencies are the sum-and-difference intermodulation products, 1.2 and 1.4 GHz. **Figure 4** shows the input-input and output-input isolation characteristics. **Figure 5** gives the variation of the input and output VSWR as a function of the input frequency. (DI #2136)

EDN

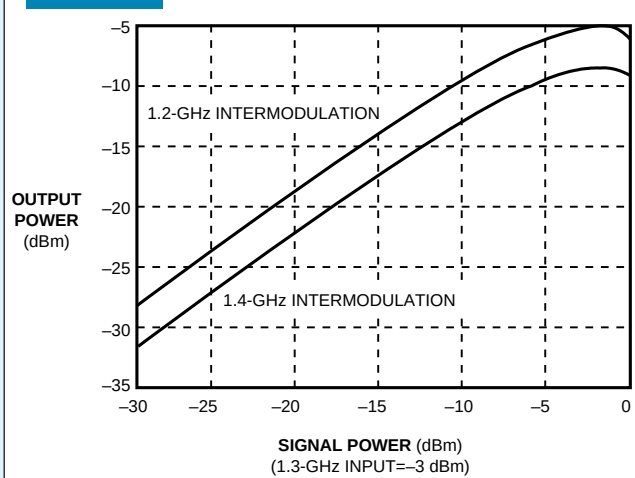
To Vote For This Design, Circle No. 328

FIGURE 2



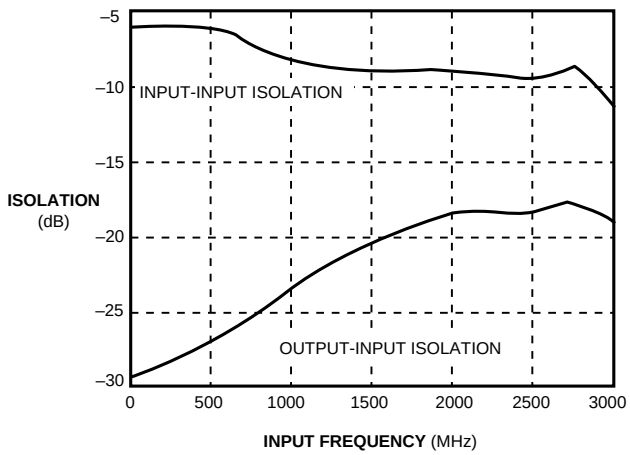
The circuit in Figure 1b is the mixer block in this 1-GHz ASK modulator.

FIGURE 3



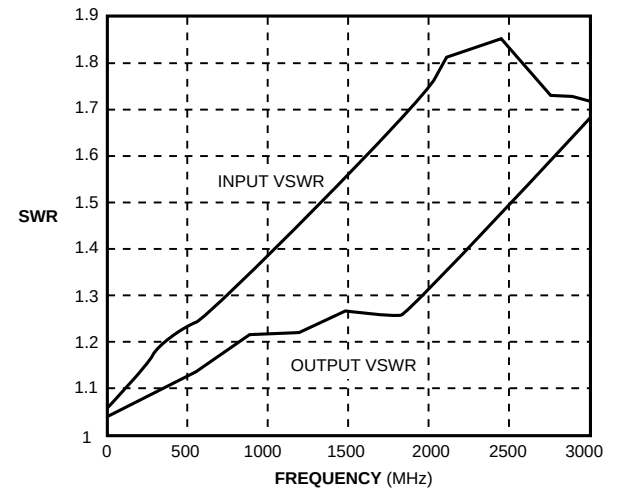
The output levels of the sum-and-difference intermodulation products in Figure 2's circuit vary linearly with the 100-MHz signal power.

FIGURE 4



Input-input isolation for Figure 2's circuit improves with increasing input frequency; output-input isolation degrades but remains reasonably high to 3 GHz.

FIGURE 5



The slight input-impedance mismatch and varying input reactance of the MMIC produces input VSWR degradation with frequency, but the VSWR remains within a respectable 1.4 at frequencies to 1 GHz.

Signal conditioning precisely indicates humidity

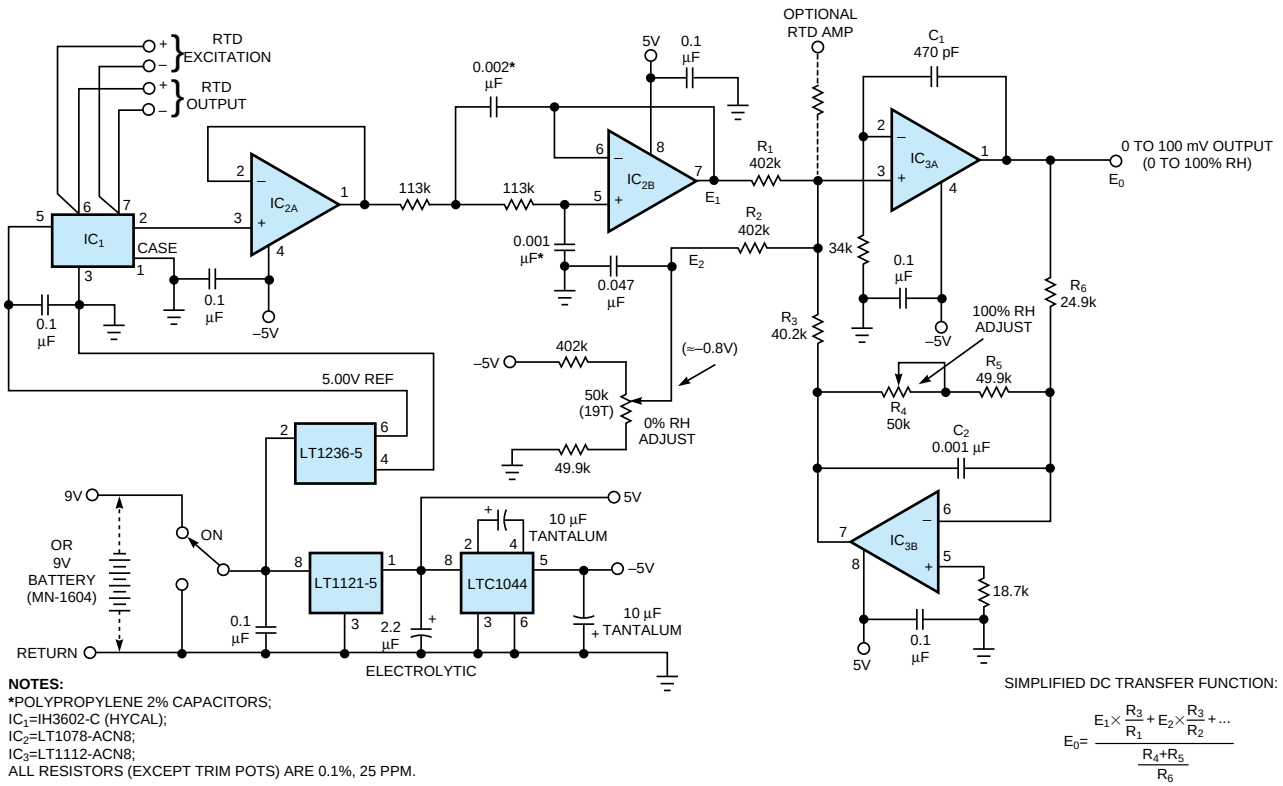
WILLIAM WHITEHEAD, LAFAYETTE, CO

The circuit in **Figure 1** translates the level of humidity from 0 to 100% into a stable, respective dc signal of 0 to 100 mV. The heart of the circuit is IC_1, the IH3602-C relative-humidity sensor (Microswitch/HyCal Sensing Products, Freeport, IL). This device operates on 5V at 200 μA and provides a dc output of 0.8 to 4.0V over the 0 to 100% humidity range. The output impedance is 5 μA when sourcing, and on-chip circuitry preconditions the humidity-related charge of a thermoset polymer dielectric capacitor. The dc output contains a small amount of 1-kHz modulation, which is an artifact of typical switched-capacitor circuits. Device accuracy is ±2% relative humidity, and linearity is ±0.5%. The six-pin TO-5 package includes a precision thin-film, 1-kΩ RTD that you can use in applications that require temperature-corrected relative-humidity data.

The LT1236-5 provides a stable 5V supply for IC_1, which precludes any ratiometric changes in the sensor's output that would otherwise occur with less stable 5V supplies. Amplifier IC_{2A}'s voltage follower buffers the sensor's high-impedance output. IC_{2B}, a 1-kHz, two-pole Butterworth filter, reduces the 1-kHz chopper modulation. The unique output stage comprising IC_3 is a precise, dc-accurate, ultralinear, noninverting summing/scaling amplifier. The LT1112 performs well for this function because it has high large-signal gain (A_{VOL}), low input bias current (I_B), and low input offset voltage (V_{OS}). The summing node of IC_{3A} is a handy spot to add a temperature-correcting term, as the **figure** indicates. C_1 and C_2 roll off any high-frequency gain/phase-related oscillations.

The 0% relative-humidity adjustment not only takes care

FIGURE 1



Precise signal conditioning transforms the 0.8 to 4.0V output of an accurate 0 to 100% relative-humidity sensor, IC_1, to a 0- to 100-mV output.

of the nonzero output of IC_1, but also compensates for any residual offsets. You achieve a full-scale output of 100 mV using the 100% adjustment.

The circuit draws 2.56 mA, which, if battery-powered, results in a battery life of about 250 hours. You can drop the total consumption to approximately 625 μA by changing IC_2 and IC_3 to LT1078s and by powering IC_1 from the 5V output of the LT1121-5. This change lengthens battery life to more than 1000 hours. However, these changes also reduce the overall accuracy of the circuit and prevent IC_3 from driving anything but very light capacitive loads, such as high-impedance ADC inputs. (DI #2145) EDN

To Vote For This Design, Circle No. 329

Vital-signs monitor consumes less than 50 μA

LEONARD SCHUPAK, DISCOVISION ASSOCIATES, IRVINE, CA

A remote data-acquisition circuit for monitoring a patient's vital signs—pulse rate, respiration, and temperature—uses very little power as well as inexpensive sensors and circuits (**Figure 1**). The data-acquisition portion uses the low-cost CD4000 series of CMOS ICs and consumes less than 50 μA from a 3V battery. The circuit uses 50× oversampled sigma-delta ADCs with a PWM/FM system, which combines three data channels on one RF carrier.

In **Figure 1**, the pulse sensor comprises a pair of conductive plastic electrodes that monitor two EKG lead points across the patient's chest. These electrodes are similar to those used on a popular and effective pulse monitor for athletic activities. The respiration sensor is an Amp (Harrisburg, PA) DT1-028K, which consists of a piezoelectric-film element mounted on a flexible beam that attaches with a chest strap similar to the same athletic monitor. The strap uses the beam as one of its suspension loops, the electronics assembly attaches to the other end, and the complete assembly straps to the patient's chest. The temperature sensor comprises a 50-kΩ thermistor and linearizing resistor mounted in the EKG electrode, which makes a good thermal connection to the patient's body.

Using the unique properties of the CD4069 as an analog component, the circuit implements two second-order sigma-delta ADCs that sample the ac components of the pulse (EKG) and respiration at a nominal 2-kHz clock rate. The circuit develops this clock using the thermistor as a tuning element in a temperature-to-frequency converter. Temperature is a very low-frequency (essentially dc) data-channel component, and pulse and respiration are higher frequency ac. Thus, you can easily combine all three channels in one serial data channel for transmission over the RF link.

The analog circuit also includes an open-lead detector that rings an alarm if the EKG lead loses contact. IC_{1C} and IC_{1F} form a Schmitt trigger that disables the RF-carrier output when the sense of skin conductivity is lost.

IC_{2A} and IC_{2D} form a straightforward clock oscillator. Although easy to design, this oscillator imposes some interesting calibration problems in operation. The thermistor, R_1, and tuning capacitor, C_1, determine the frequency. For a thermistor of 11 kΩ at 98.6°F, C_1 should be 0.022 μF for a clock frequency of 2 kHz. Connecting these components to IC_{2A} and IC_{2D}, as the figure shows, provides a convenient pc-board layout with good isolation from these low-level input signals. You can use an additional resistor set to optimize the frequency range, depending on the thermistor selection. The operator usually performs a calibration cycle when you apply the unit to the patient, setting the "current reading" to a known temperature obtained directly from the patient.

TABLE 1—PHASE INCREMENTS

Channel A	Channel B	Phase increment (°)
0	0	45
0	1	90
1	0	135
1	1	180

The operation of the ADC is straightforward but has some unusual aspects. (Note that because of the way the CD4069 inverter works—used here exclusively for analog-amplifier functions—the **figure** uses a one-input NAND gate as the logic symbol to differentiate from the triangle symbol.) An ADC channel comprises three CD4069 inverters in series, for which two operate as analog integrating amplifiers biased in the linear region. In this configuration, each CD4069 section yields about 30 dB of gain and a bandwidth in excess of 2 kHz. The third inverter and flip-flop input act as the typical comparator part of the ADC.

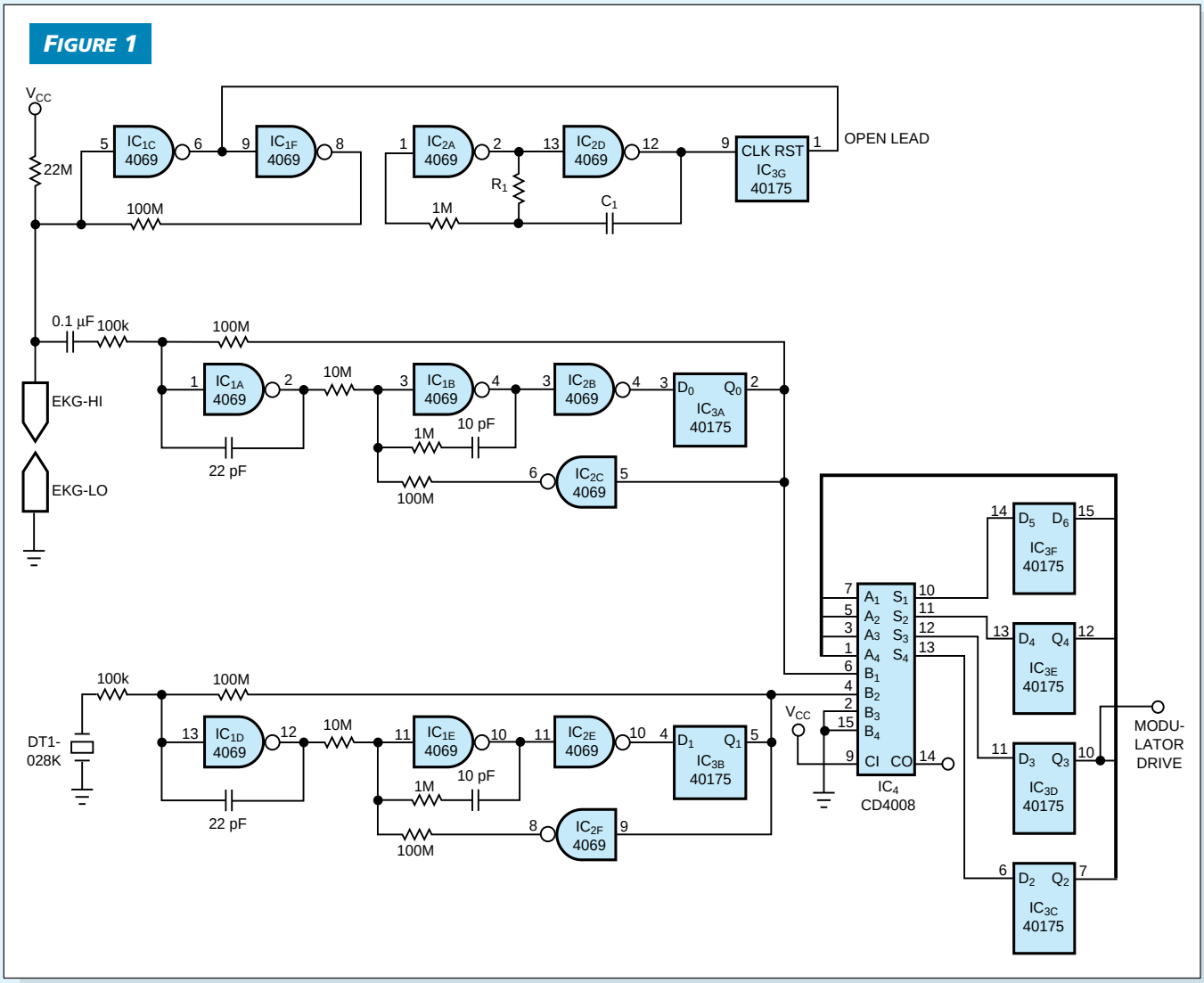
The first stage is a relatively uncomplicated integrator with an indeterminate time constant of approximately 1 to 10 μsec. The second stage has a pole/zero response that flattens to –20 dB at approximately 5 kHz. Feedback from the sampling register (flip-flop) provides a stable, wideband

inner loop as well as the main outer loop. The 100-MΩ resistors in the feedback loops are readily available from several vendors in surface-mount configuration and are reliable when you use good assembly practices.

The components in the **figure** comprise only the front-end input portion of the design; the noise filter and subsequent decoder/application circuits reside at the receiver. Before transmission, the circuit combines the three channels of data into one modulating signal using a PCM format similar to modified FM (MFM) for which the temperature-controlled clock oscillator determines the pulse-rate clock. In other words, the transmitter uses a complex set of modulating codes that include a PCM/FM component superimposed on an amplitude-shift-keyed (ASK) carrier. The temperature-sensitive oscillator frequency varies about 20% total over the expected range of body temperatures, and this variable rate is adequate to transmit the digital PCM data.

A CD4008 full adder encodes the digital data from each ADC flip-flop in MFM form. A simple 3-bit digital synthesizer accomplishes the encoding; the contents of output registers IC_{3C} and IC_{3D} determine the output phase. The ADC's data contents determine the incremental phase change at each clock (**Table 1**). The circuit then uses the most significant bit of the output register as the ASK signal. The 4-bit register then stores the sum of the data, or the phase increment with the current contents, representing the digital phase of the modulation carrier. The maximum resulting fundamental frequency of this carrier is then one-half the clock frequency. (DI #2149) EDN

FIGURE 1



Inexpensive sensors and simple CD4069-based sigma-delta ADCs comprise the front end of low-power, remote data-acquisition for monitoring a patient's temperature, pulse, and respiration.

Optocoupled gate detects motor operation

JIM KNIPFER, LIEBEL-FLARSHEIM CO, CINCINNATI, OH

Most applications require redundant system checks to ensure that devices are operating as expected. The circuit in **Figure 1** detects when a PWM-controlled servo motor is running. You can monitor the motor-running signal output with a CPU or tie the output to hardware that indicates a fault if the motor is running when it should be off. The motor's output connects to input resistors R_{IN1} and R_{IN2}. The use of two resistors protects the motor or servo amplifier in the case of a short circuit. R_{IN1} and R_{IN2} also serve as current-limiting resistors for the LED in the opto gate, IC_2.

The value of R_{IN1} and R_{IN2} in **Figure 1** is suitable for 150V nominal servo output voltage. Because the input voltage can reverse, the circuit uses a full-wave bridge (D_1 to D_4) before the input of the opto-gate IC. The circuit needs a high-speed opto gate, because when the motor runs at low speeds, the servo-output duty cycle can be only a fractional percentage, thereby providing only 1-μsec or narrower pulses to sample. When no signal is present at the input resistors, the output of the opto gate is high, causing the output of inverter IC_3 to be low. The low output allows the MC14541 counter, IC_1, to count at the frequency determined by R_1, R_2, and C_1.

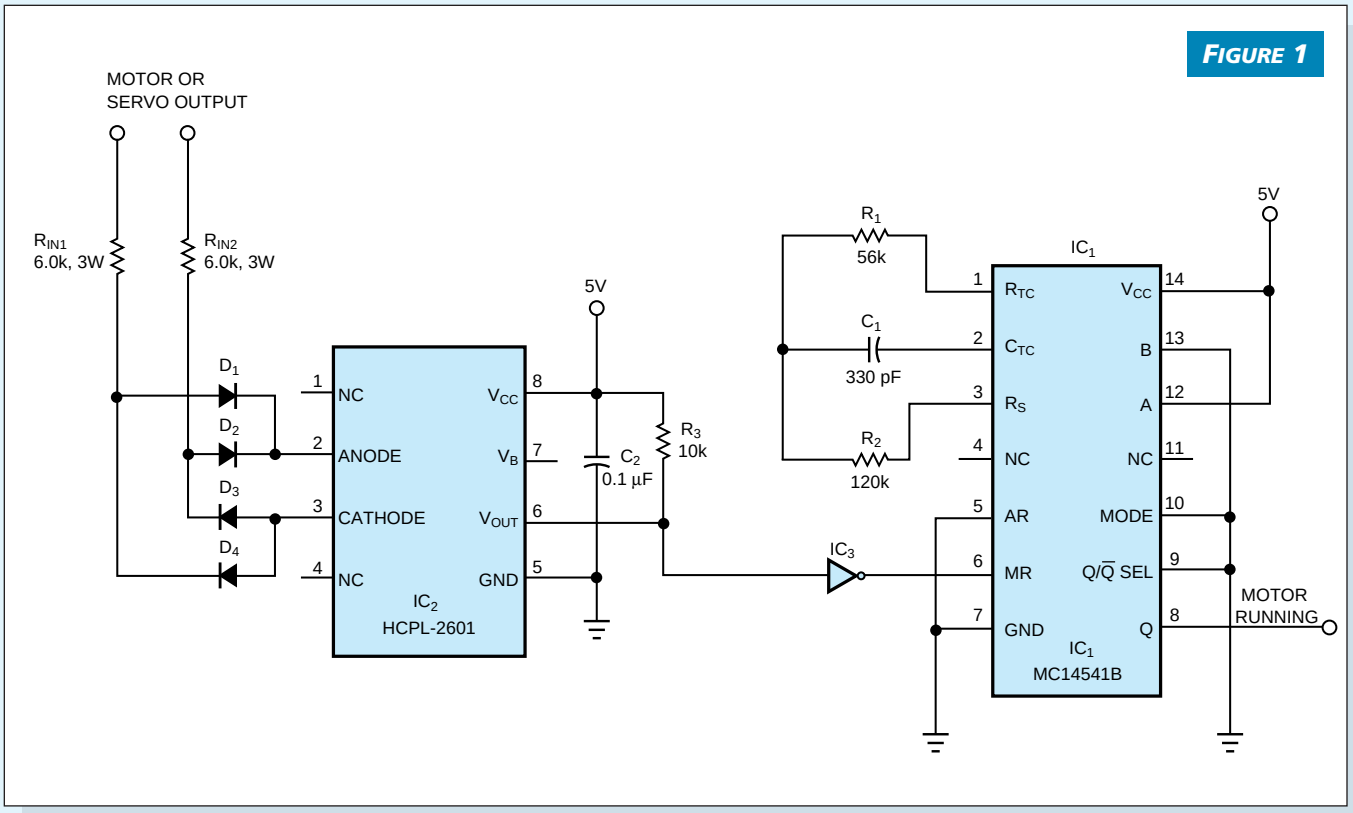
TABLE 1—APPROXIMATE TIME-OUT PERIODS

A	B	Approximate time-out period
0	0	350 msec
0	1	45 msec
1	0	10 msec
1	1	2.8 sec

You can determine the time-out by multiplying the count range (selected by inputs A and B) by the period of the frequency set in IC_1. **Table 1** shows approximate time-outs. Upon the initial application of power, the motor-running signal remains active until the time-out occurs. Upon detection of a servo pulse of 500 nsec or longer, the master reset pin (Pin 6 of IC_1) goes high, resets the internal counter, and asserts the motor-running signal. The motor-running signal remains low for the amount of time shown in **Table 1**. The time-out for the circuit in **Figure 1** is approximately 10 msec. (DI #2155)

EDN

FIGURE 1



Did you leave your motor running? This circuit lets you or your μC know.

Scheme implements multiple output ports

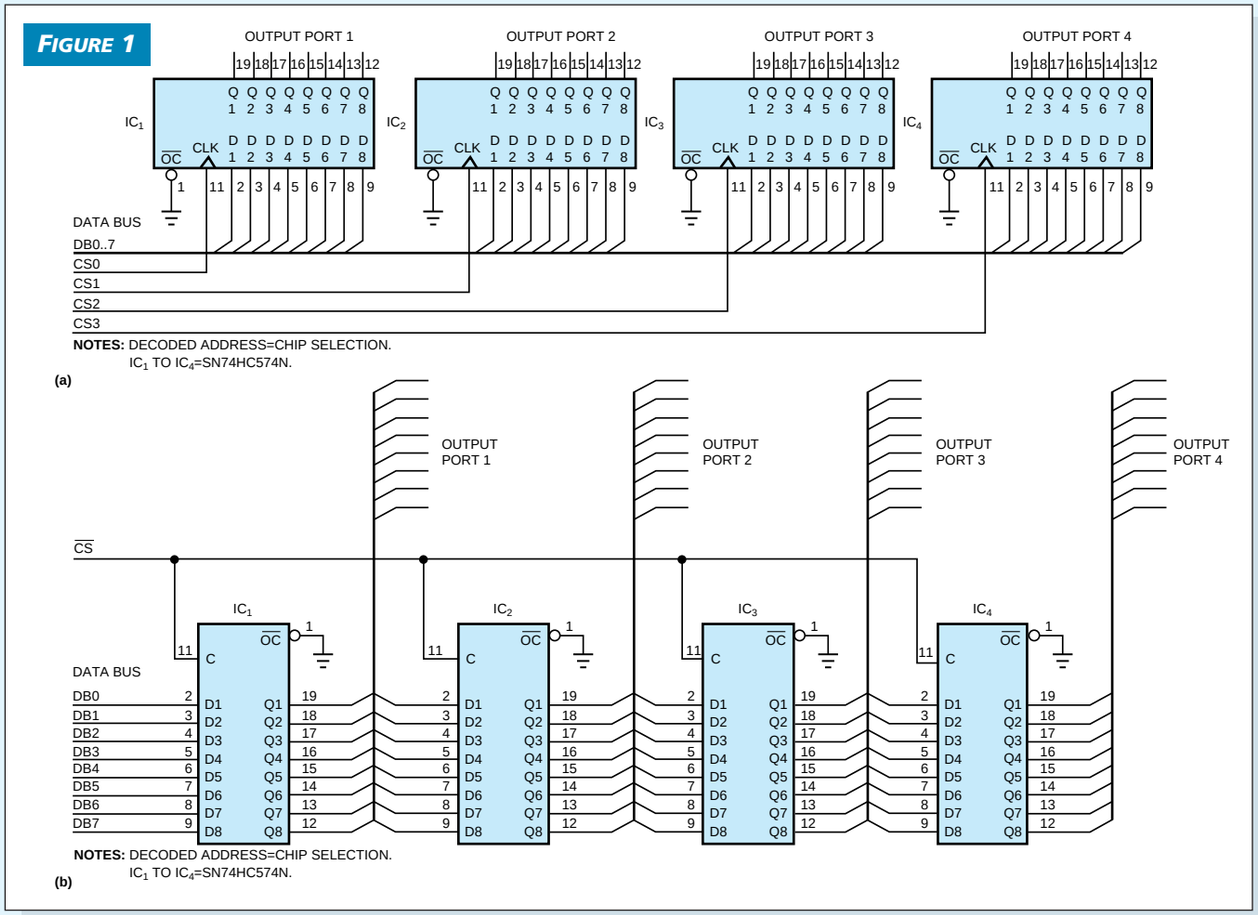
W KURDTHONGMEE, NAKORN SI THAMMARAT, THAILAND

In μP/μC systems, you sometimes need to use many output ports and update all the ports simultaneously. For instance, in systems that use multiplexing techniques, the output ports require refreshing for every scanning period. One example is a system of multiple dot-matrix LEDs. Suppose that an N-column, 8×8 LED matrix uses a row-scanning technique. Every eight-dot column of the n^{th} LED connects to the n^{th} port (n=1…N) and the m^{th} row (m=0…7), in which row a switching transistor controls all the LEDs. To update each row, the row's switching transistor turns off, and all N ports receive an update from the display buffer. The switching transistor for that row then turns on. The method in **Listings 1** and **2** and **Figure 1** allows you to simultaneously update all N output ports.

Two approaches are available for implementing the parallel-output port. In the first approach, every output port has its own address. The address-enable lines of the output ports connect to the chip decoder. To update the output ports, the application program needs to directly refer to the port's address (**Figure 1a**). The native MCS-51 assembly routine in **Listing 1** simultaneously updates all N output ports in the configuration of **Figure 1**. Assume the addresses of these ports are OUTPUT_PORT0, OUTPUT_PORT1, OUTPUT_PORT1+1…OUTPUT_PORT(N–1).

Alternatively, you can connect all the output ports in the configuration of **Figure 1b**. In this method, only one address refers to all the output ports. For example, to update all N ports, the application program must refer to the ports' address and then iterate the following steps N times: Read data from address N from the buffer, route the data to the

FIGURE 1



One way to update output ports is to address every port (a); another method is to address them all simultaneously (b).

referred port, and let N=N–1.

Note that, in the above example, data from location n in the buffer updates the n^{th} port, where n is between 0 and N–1. **Listing 2** is the MCS-51 native routine to simultaneously update the output ports. Assume that the single address of the N ports is OUTPUT_PORT. By using this single-address configuration, you do not need to refer to output ports' addresses every time you need to update them. You thus eliminate one instruction in the update loop, making the routine shorter and faster. This routine is suitable for implementation in an interrupt-service routine. (DI #2151)

e

To Vote For This Design, Circle No. 381

LISTING 1—MCS-51 ROUTINE TO UPDATE OUTPUT PORTS

```
update_1:  mov   dptr,#OUTPUT_PORT0     ; Offset of output port
           mov   r0,#buffer             ; Offset of buffer
           mov   r1,#00h                ; Loop control
update_1:  mov   a,@r0                  ; Read from buffer
           movx  @dptr,a                ; Update current output port
           inc   dptr                   ; Next output port
           inc   r0                     ; Next location in buffer
           inc   r1
           cjne  r1,#N,update_1         ; Until N ports
           ret
```

LISTING 2—MCS-51 ROUTINE FOR SIMULTANEOUS UPDATE

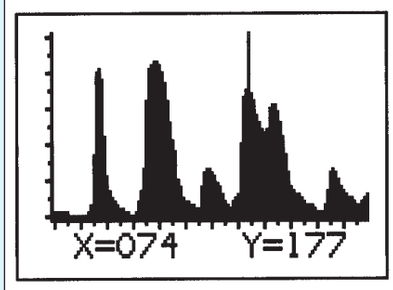
```
update_2:  mov   dptr,#OUTPUT_PORT     ; Point to output port address
           mov   r0,#buffer            ; Offset of buffer
           mov   r1,#00h               ; Loop control
update_1:  mov   a,@r0                 ; Read from buffer
           movx  @dptr,a               ; Update current output port
           inc   r0                    ; Next location in buffer
           inc   r1
           cjne  r1,#N,update_1        ; Until N ports
           ret
```

PIC plots pixels sans controller

DARYL CHRISTOPHER, JUNIATA COLLEGE, AND TOM FISHER, INEXPENSIVE SYSTEMS, HUNTINGDON, PA

It is sometimes desirable to display the output of a PIC μC graphically, rather than numerically. However, the operation increases system complexity and expense. In addition to the LCD itself, you usually need a graphics controller and at least 1 kbyte of RAM. However, you can obtain a satisfactory plot (**Figure 1**) without the use of a controller or any RAM. At first glance, the parameters seem daunting. The least expensive graphics LCDs (for example, the Optrex DMF696, at $39.51 from Digikey (Thief River Falls, MN)) have a resolution of 128×64 (width×height) pixels, which equates to 1024 bytes. A midrange PIC μC, such as the PIC 16C37A, has only 192 bytes of RAM available, including whatever you need for program variables.

FIGURE 1



This simulated plot shows the blinking cursor at X=74 measuring Y=177.

Furthermore, you must frequently refresh the LCD: Optrex recommends a 70-Hz rate. The PIC's cycle time (the time it takes the μC to execute one command) is 200 nsec. The μC divides a 20-MHz oscillator input by 4 to produce the 5-MHz cycle frequency. Because the LCD needs 8192 pixels at an approximate 70-MHz refresh rate, the PIC has the time to issue only approximately nine commands for each pixel. Moreover, this time must include the timing pulses (**Figure 2**) as well as the data.

The key to matching the PIC's capabilities to the task is recognizing that a graph is a specialized display. The plot is of a single-valued function of y vs x, so the display has 128 points at any time. The majority of the pixels carry no information whatsoever. Inasmuch as the maximum resolution possible on the vertical axis is only one part in 64, 1 byte (1-in-256 resolution) is more than adequate to contain one datum, and the 192 bytes of PIC RAM is more than enough for the plot. (To allow for borders, the graph in **Figure 1** actually has a plottable area of only 120×53 pixels.)

Another problem is to write code that is terse enough to maintain the required refresh rate. The fastest way to *move* a bit (pixel) out of the PIC is to rotate it through the carry flag

and into an output port. This operation would normally mean wasting an entire output port, because the remainder of the port's pins would always contain "stale" shifted pixels. However, the PIC allows A/D input through pins designated for digital input, so all the I/O pins are usable. **Figure 3** shows the PIC 16C73 port assignments. **Listing 1** shows the code fragment that displays a single pixel on the screen.

Note that the routine requires only six cycles. However, if you include the start-of-line pulse (CP1, two cycles), subroutine calls (two cycles), and other overhead, the worst case requires 12 cycles. This time budget results in an unacceptable refresh rate of approximately 50 Hz. The screen areas that require unchanging information, such as borders, tick marks, and labels, are stored in program ROM. Because it would take too many cycles to recover this data from look-up tables (which are cumbersome in the PIC), you code each byte explicitly and clock them out 1 bit at a time using the routine in **Listing 2**.

You need approximately 700 lines of repetitious coding and a large, binary shape table to specify the static information; a good editor simplifies this chore. You implement a blinking x-axis cursor by periodically replacing the datum in the appropriate RAM location with 255 decimal. This operation has the effect of drawing a flashing vertical line above the pointed-to datum. By setting 64 screen scans with and 64 without the 255, you obtain a blink rate of approximately 2 sec. You control the cursor's position by rotating a 50-kΩ potentiometer that feeds the PIC's A/D converter through Pin RA2. As the cursor moves, the x and y coordinates of its position display numerically at the bottom of the graph. The updatable, three-digit x and y positions (requiring a total of 6 bytes of RAM) make up a classic character generator, with each numeral's shape stored in 7 bytes of RAM.

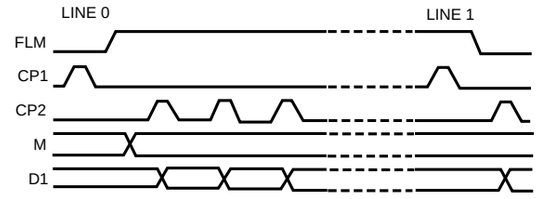
LISTING 1—CODE FRAGMENT FOR A SINGLE PIXEL

```
incf    FSR         ;Point the indirect address vector to the
                     next datum to be displayed.
movfw   IndF        ;Get that datum and place it in the working
                     register.
subwf   Ylevel      ;Compare the value of the datum to the height
                     of the scan line currently being written to
                     the screen. If the datum is greater, set the
                     Carry Flag.
bsf     PortC,CP2   ;Set the latch pulse so that the I/O pin
                     can change state.
rlf     PortA       ;Place the pixel information on the output pin.
bcf     PortC,CP2   ;Latch the pixel and send it to the LCD.
```

LISTING 2—CODE FOR OBTAINING STATIC INFORMATION

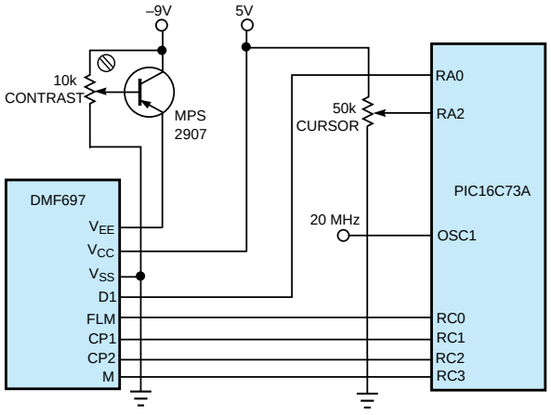
```
movlw   B'00000111'     ;Get the bit pattern of a portion
                         of the left border.
movwf   Byte            ;Store it in the pixel buffer.
call    Write_Text_Byte ;Send the byte to the LCD one bit
                         (pixel) at a time.
```

FIGURE 2



The 16C73A sends these signals to the DMF697. FLM is start-of-scan, CP1 is start-of-line, CP2 is data latch, M is driver-voltage polarity, and D1 is data.

FIGURE 3



This minimal circuit displays the graph and permits cursor movement; it makes no provision for acquiring data.

It is essential to frequently update the screen to continuously reverse the polarity of the LCD's electrodes and thus prevent the destruction of the display. The M line takes care of this operation by switching the polarity at the beginning of each screen scan. Although it is possible to invoke short routines between scans, if the PIC must perform lengthy routines, it is necessary to blank the screen by using a call to Erase_Screen. The program requires approximately 1.4 kbytes of program memory (in Page 1 ROM) and 135 bytes of data RAM. You can download the program as an MPASM 1.40-compatible source file from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2153. Although using a PIC μC as an LCD controller/video RAM imposes severe restraints on what you can display, this approach reduces complexity and cost for simple graphs. (DI #2153) e

To Vote For This Design, Circle No. 382

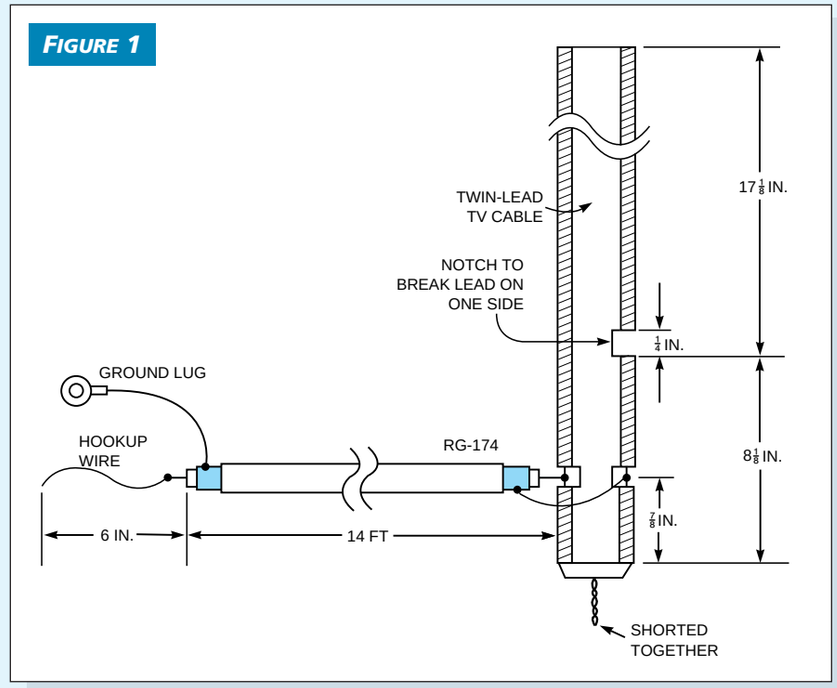
Antenna extension provides open-door policy

RICHARD PANOSH, VISTA, BOLINGBROOK, IL

Metal garage doors may be solid and secure, but they are not transparent to radio waves. If your garage has aluminum siding, you may experience frustration with the operation of your garage-door opener. If you must drive up to the door and nearly touch it or drive near a window to activate the door opener, this Design Idea is for you. The antenna extension in **Figure 1** moves the receiving antenna beyond the door so the system can respond to your command.

Construction is simple. The antenna is completely passive and uses a J-pole design. The antenna uses readily available 300Ω flat TV twin-lead and RG-174 50Ω miniature coaxial cable. The length suits an opener that operates at 310 MHz. Most popular openers, such as Stanley, Genie, and Chamberlain, operate at this frequency. However, some units operate at 315 MHz, some replacement controls operate at 390 MHz, and some automobile manufacturers offer 380-MHz designs. You can usually find the operating frequency in the user's manual, or you can obtain it directly from the manufacturer.

FIGURE 1



A sensitivity treatment for your garage-door opener uses readily available coaxial cable and TV twin-lead in a simple configuration.

A J-pole antenna consists of a half-wave antenna matched to the low impedance of the coaxial cable by means of a quarter-wave matching stub. In **Figure 1**, the half-wave antenna comprises the piece of wire above the notch cut into one side of the 300Ω flat TV twin-lead cable. You solder the lower ends of the stub together. Experiments show that the optimum location of the coaxial tap is 7/8 in. above the shorted end. You use 14 ft of RG-174 coaxial cable as the lead-in. Route this lead above the garage door and out through the weather stripping without drilling any holes. Be sure that the center conductor of the coaxial cable connects to the half-wave side of the twin-lead and that the ground braid connects to the notched side of the twin-lead. Solder both these connections and cover them with clear silicone rubber to protect them from weather conditions.

Prepare the end of the coaxial cable that connects to the existing garage-door antenna by exposing the center conductor and soldering on a 6-in. length of hookup wire. You can insulate the connection with heat-shrink tubing or tape. Twist the hookup wire tightly around the existing antenna wire for its full length to form a gimmick capacitor suitable for coupling the signal into the receiver without unduly loading the receiver and altering its characteristics. Secure the shield of the RG-174 coaxial cable to the frame of the door opener to effectively ground it. You can locate the antenna above the garage door on the wood trim. Do not staple across the 300Ω twin-lead.

Mount the twin-lead to the door using adhesive pads and nylon cable ties or by means of a single small nail or screw at each end through the center web of the twin-lead. You can attach the coaxial cable to the garage rafters or ceiling with the same nylon cable ties and adhesive to keep it out of the way of moving parts. Route the cable above the door to allow normal operation of the door and out below the garage-door header beam so that the door's weather stripping seals when the door closes.

The addition of this antenna extension to any garage-door opener markedly increases operating range. Measurements show an improvement of approximately 2.5 dB, equivalent to a 30 to 40% increase in range. The improvement is especially noticeable for installations in which the garage is effectively shielded by metal doors, metal siding, and a lack of windows. (DI #2152) e

To Vote For This Design, Circle No. 383

Dual comparators stabilize proximity detector

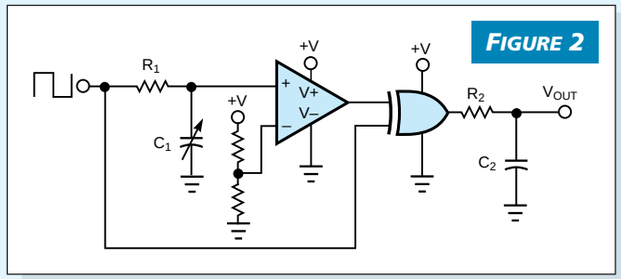
ARTHUR HARRISON, US ARMY RESEARCH LABORATORY, ADELPHI, MD, AND JOSEPH STERN, MAXIM INTEGRATED PRODUCTS, SUNNYVALE, CA

In the proximity detector of **Figure 1**, a 4-in.-sq piece of copper-plated pc board serves as an antenna that forms one plate of a capacitor. An approaching (grounded) person serves as the other plate, producing a capacitance value of 2 to 5 pF that increases as the person approaches. At 6 in. from the copper plate, for example, the person produces a capacitance value of approximately 2 pF.

A simplified circuit illustrates how the circuit transforms distance/capacitance into a proportional voltage (**Figure 2**). Transitions of the input square wave apply directly to the lower input of the exclusive-OR (XOR) gate but are delayed $0.693 \times R_1 \times C_1$ sec before the comparator reconstructs the transitions and applies them to the upper XOR input. R_2 and C_2 filter the resulting XOR output to produce a voltage proportional to distance.

The XOR output's duty cycle is proportional to the sum of the R_1-C_1 network's delay and the comparator's propagation delay, so a small variation in comparator delay can mask small changes in antenna capacitance. The circuit in **Figure 1** overcomes this limitation using a dual comparator (IC_1). Passing the XOR inputs through nearly identical comparators largely nullifies the effect of offset voltage, drift, and propagation delay through the comparators.

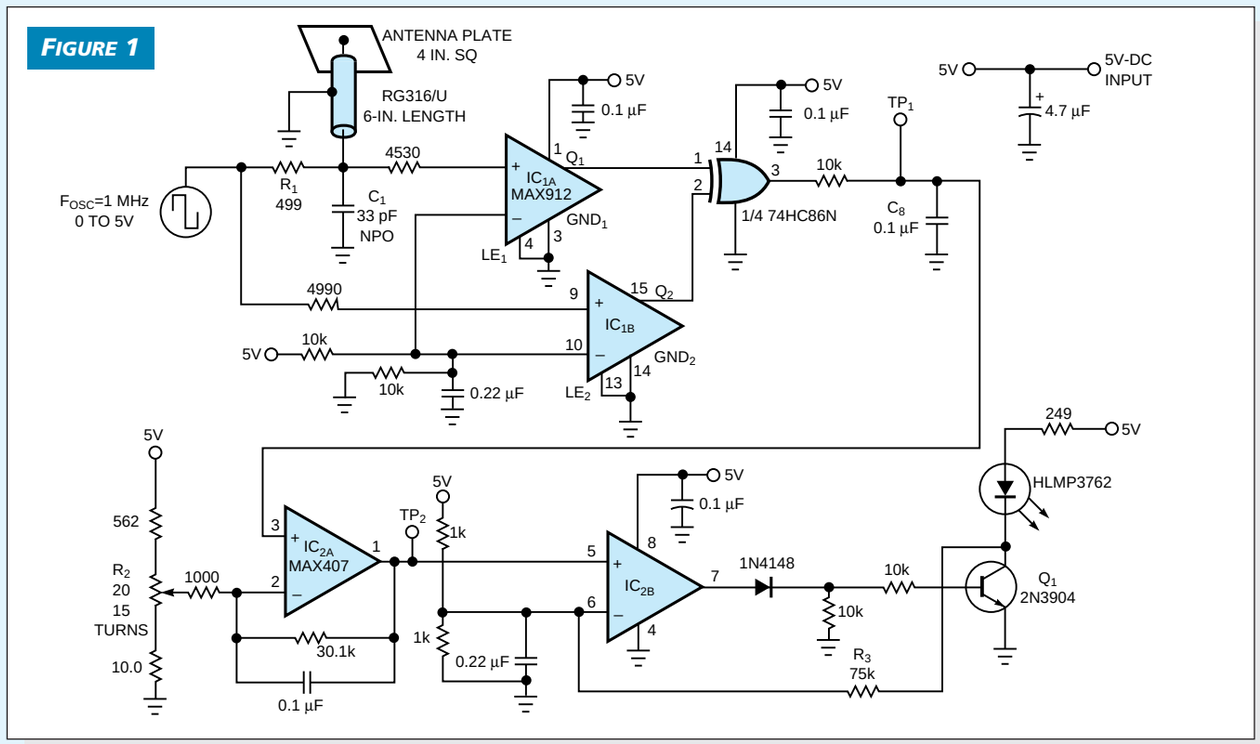
FIGURE 2



Exclusive-ORing two inputs, one delayed by the R_1-C_1 network, and subsequent filtering by R_2 and C_2 implements a capacitance-to-voltage conversion.

Figure 1's delay capacitance consists of a 33-pF capacitor, C_1, in parallel with 15 pF (6 in. of coaxial cable at 30 pF per foot) and the 4-in.-sq antenna plate. This capacitance charges to 5V through R_1 during each positive half cycle of the input square wave. When no one is near the detector, this capacitance equals 48 pF and produces a delay of 16.5 nsec at the upper XOR input. With a hand 6 in. from the detector, the capacitance rises to 50 pF and produces a delay

FIGURE 1



This proximity detector lights the LED when a person approaches the antenna plate within a threshold distance set by potentiometer R_2.

of 17.3 nsec, yielding a time difference of only 0.8 nsec.

To detect such small time differences—over temperature and with accuracy—the comparators must be stable in offset voltage and propagation delay. (Changes in offset voltage as well as propagation delay affect delay time.) One 10-nsec comparator is generally stable to within 1 nsec. To resolve subnanosecond intervals, use the dual-comparator approach of **Figure 1**, which increases the useful resolution by a factor of four to five.

Op amp IC_{2A} offsets and amplifies the dc voltage at TP_1, which corresponds to the distance between hand and antenna plate. A hand movement toward the antenna causes the voltages at TP_1 and TP_2 to rise. IC_{2B} and Q_1 serve as a comparator with hysteresis, which compares the TP_2 voltage with 2.5V. Thus, any TP_2 voltage above 2.5V (which corresponds to a proximity of 6 in.) turns on the LED. You can adjust potentiometer R_2 to set a threshold other than 6 in., and you can connect a DVM at TP_2 to read out the proximity in inches. R_3 adds hysteresis to ensure a well-defined transition.

To ensure frequency stability for the high-speed dual comparator in **Figure 1**, the copper-clad pc board should have a ground layer in addition to the circuit layer. Power-supply bypassing should include 0.1-µF ceramic capacitors that sit very close to the comparators' supply terminals. (DI #2150) e

To Vote For This Design, Circle No. 384

Low-cost switcher converts 5 to 24V

PAUL C FLORIAN, PLANO, TX

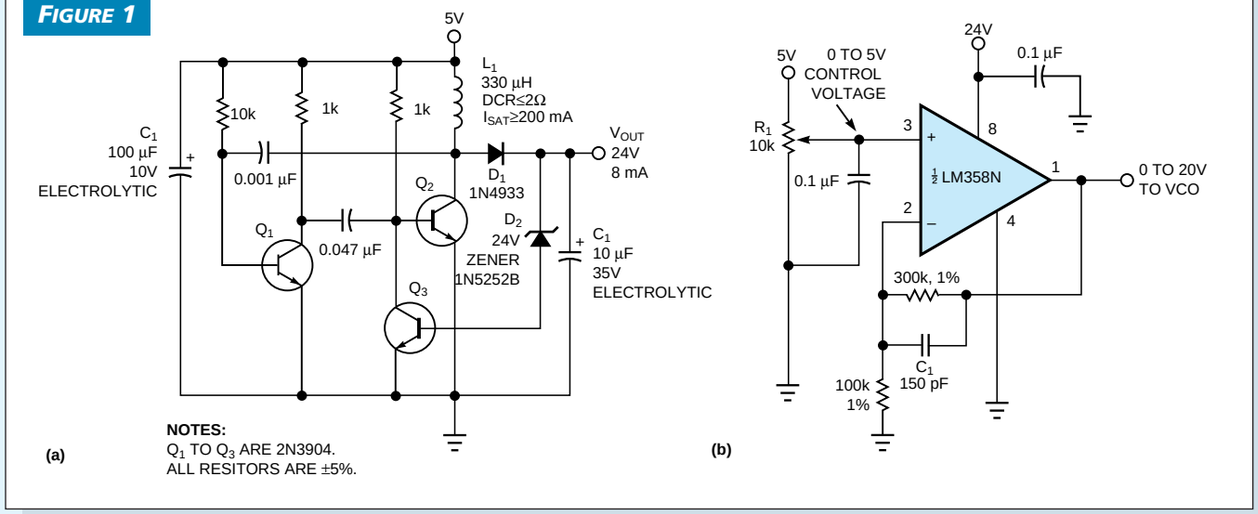
The low-cost, three-transistor boost switching regulator in **Figure 1a** is a modified astable multivibrator comprising Q_1, Q_2, and L_1, which substitutes as a load for Q_2. At the full output power of 200 mW, the oscillation frequency is approximately 60 kHz. The efficiency is 65% with V_{OUT} equal to 24V and sourcing 8 mA.

When the base of Q_2 is high, energy stores in L_1's magnetic field. When the circuit drives the base of Q_2 low, the induced voltage from L_1's magnetic field collapses to add with the supply voltage. This voltage spike charges C_1 through D_1. When the accumulated charge in C_1 results in a voltage equal to the zener voltage of D_2 plus 0.6V, Q_3 pulls Q_2's base to ground, decreasing the amount of time Q_2 is on in subsequent oscillations and thereby decreasing the energy transferred to C_1. This feedback through D_2 regulates the output voltage to 24.6V±the tolerance of D_2. To change the output voltage of the circuit, simply change the zener voltage of D_2.

Many VCOs require tuning voltages as high as 20V, and you can use this switching regulator to generate a 0 to 20V tuning voltage from a 0 to 5V control voltage (**Figure 1b**). The circuit configures one-half an LM358N as a noninverting amplifier with a gain of 4. C_1 eliminates gain for the noise generated by the 24V supply. You can manually adjust the tuning voltage using R_1 or control the voltage using feedback from a PLL. (DI #2159) e

To Vote For This Design, Circle No. 385

FIGURE 1



A simple three-transistor switching regulator (a) supplies 24V and 8 mA. The circuit can help provide a 0 to 20V VCO tuning voltage from a 0 to 5V control voltage (b).

60-Hz modulator records process variables

WARREN JOCHEM, RESEARCH TRIANGLE INSTITUTE, RESEARCH TRIANGLE PARK, NC

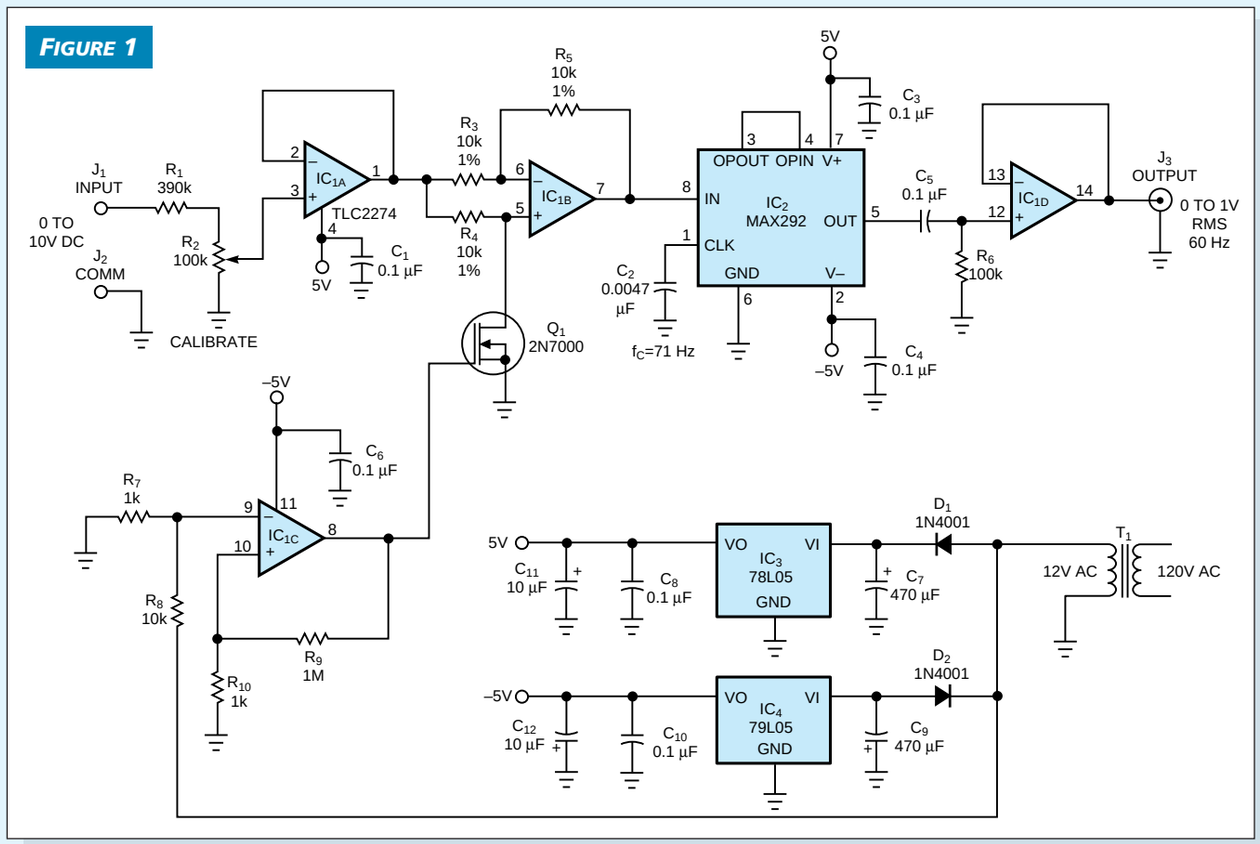
The circuit in **Figure 1** allows you to record process variables (4 to 20 mA, 0 to 10V dc) on a three-phase power monitor designed to record only ac waveforms. Many of these recorders have a seventh channel, normally used for recording neutral current, which you can use as a process-variable input. The circuit operates by generating a 0 to 1V-rms output sine wave whose amplitude is a function of a 0 to 10V-dc input signal. IC_1 can be any rail-to-rail quad op amp rated for ±5V power supplies. Input stage IC_{1A} buffers the input-voltage divider R_1-R_2 and drives the two-quadrant multiplier, IC_{1B}. IC_{1B} acts as a multiplier by using Q_1 to switch its gain from 1 to –1.

When Q_1 is off, IC_{1B} has a gain of –1. Switching Q_1 at 60 Hz chops the dc signal applied to the input into a 60-Hz square wave whose amplitude is proportional to the input signal. The chopping signal comes from Schmitt trigger IC_{1C}. A portion of the ac power signal goes to IC_{1C} through voltage divider R_8-R_7. R_9 and R_{10} provide a small amount of hysteresis to prevent oscillation and ensure fast switching. The output of IC_{1C} is the 60-Hz square wave that controls Q_1. The chopped signal from IC_{1B} connects to the switched-capacitor filter, IC_2. This eight-pole lowpass filter converts the square wave from IC_{1B} to a sine wave.

Capacitor C_2 sets the filter's 71-Hz cutoff frequency. C_5 and R_6 form a 16-Hz highpass filter that removes any dc offset from the output of the switched-capacitor filter. IC_{1D} buffers the filter and provides a 0 to 1V-rms, 60-Hz output. Calibration involves applying a 10V-dc signal to the input and adjusting R_2 until the output reads 1V rms on an ac voltmeter. Measurements show linearity within ±1% over the entire input range. To use the recorder, connect the input to the process variable under measurement, and connect the output to any voltage-input channel on the 1V-rms power recorder. You can also use the circuit on current-input channels designed to use low-voltage, 1V-rms current clamps. To record 4- to 20-mA signals, shunt the input with a 500Ω resistor. (DI #2157) e

To Vote For This Design, Circle No. 386

FIGURE 1



Chopper techniques convert a dc input signal from a process variable to a 60-Hz ac signal for measurement on a three-phase power monitor.

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____ Fax _____

E-mail _____

My e-mail address may be published Yes ____ No ___

Company _____

_____

Address _____

_____

Country _____ ZIP _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

The winning Design Idea for the September 1, 1997, issue is entitled "Transistor trio makes vector anemometer," submitted by W Stephen Woodward of University of North Carolina (Chapel Hill, NC).

Synchronizing controller detects baud rate

WILLIAM GRILL, RIVERHEAD SYSTEMS, LITTLETON, CO

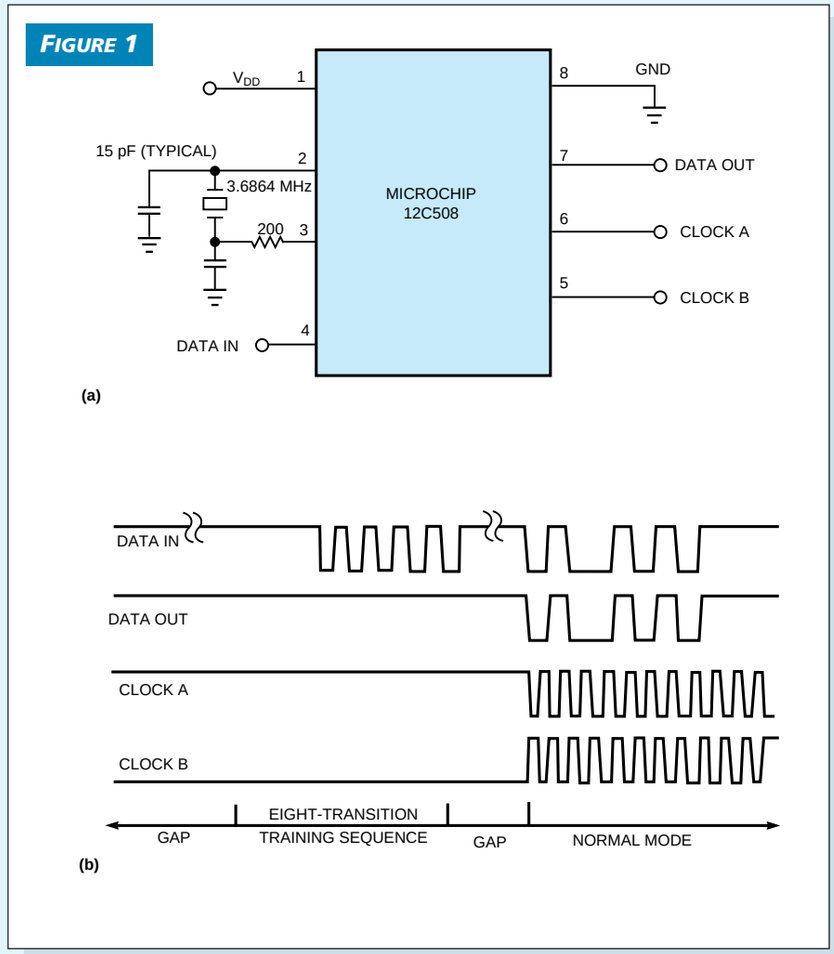
A simple and inexpensive implementation using an eight-pin 12C508 controller (Microchip Technology, Chandler, AZ) provides both bit-rate detection and a synchronous, appended-clock output from an asynchronous input-data stream (Figure 1a).

This implementation relies on two code sequences (Figure 1b). A training sequence first identifies the bit rate. This sequence begins with register and counter initialization followed by a gap detection. Following the detected, 0.325-sec, high-true gap, the controller monitors eight transitions. Between each transition, a local loop counter counts the number of loops necessary to arrive at the next transition. The controller tests the accumulated number of loops between transitions to identify the count period that resulted in the fewest loops. After processing all eight transitions of the training sequence, the controller evaluates the shortest maintained count according to a table index based on the training loop s code length. Because the controller s timing is crystal-based and the loop s path lengths are equal, the processed count corresponds to the bit rate that the controller then uses to define a delay necessary for the second code sequence.

The second output-code sequence begins by locating a fixed, second gap of approximately 40 msec. This time allows the controller to completely terminate the training sequence before beginning the controller s main iterative loop. The sequence then takes the delay, identified earlier, and processes the serial input data on Pin 4 to the output at Pin 7 while maintaining complementary clocks on pins 5 and 6. Equalizing the instruction paths for this code sequence is a key requirement of this application.

The controller uses a 3.6864-MHz crystal to provide precise baud-rate timing. This scheme allows the combined code applications to support 300- to 9600-bps asynchronous inputs. An 8-bit training pattern of 55 (hex) allows multiple single-bit isolated transitions to qualify the detected bit rate (Figure 1b).

The output-sequence code provides resynchronization of the clock-related delay counters at the input data-transition edges on Pin 4. This resynchronization allows flexibility in supporting data-bit edge delays and distortion and variations in the controller s crystal or external timing reference s accuracy. You can download applicable code from EDN s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2158. (DI #2158)



A simple controller (a) provides baud-rate information and a synchronous output clock by using two code sequences (b).

To Vote For This Design, Circle No. 343

Charge Li-ion batteries from ac line voltage

MATT SCHINDLER, MAXIM INTEGRATED PRODUCTS, SUNNYVALE, CA

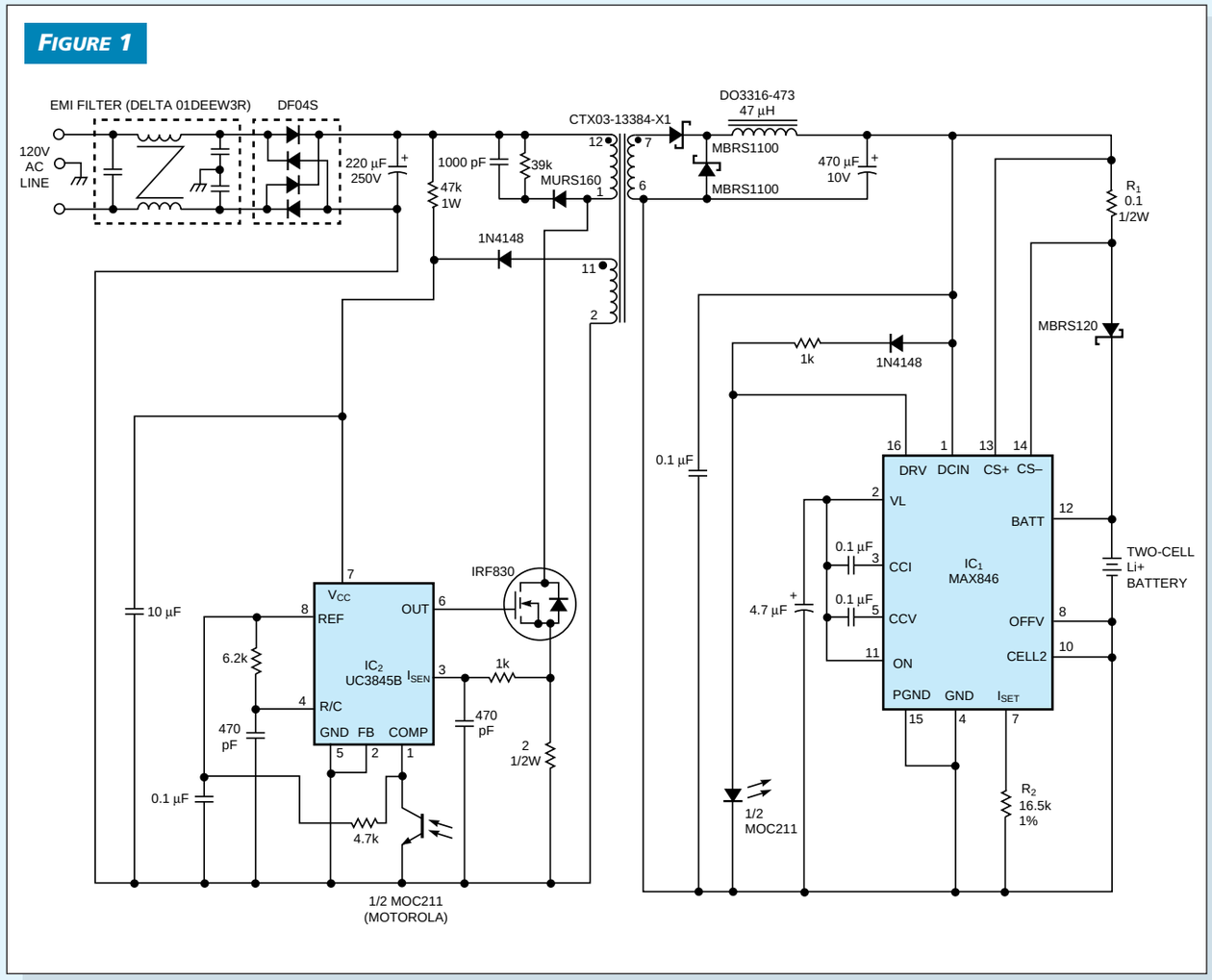
Li-ion battery chargers must apply constant current when the battery charge is low and constant voltage when the battery charge is high. To avoid battery damage, the tolerance on the applied voltage must be less than 1%. The charger in Figure 1 complies with these requirements.

The circuit converts energy from 120V ac to a regulated voltage or current as necessary to charge two Li-ion cells in series. IC_1, a popular controller for offline power-supply applications, operates as a forward converter, producing an isolated, half-wave-rectified battery voltage or current from the full-wave-rectified line voltage. This converter operates at 250 kHz and handles ac inputs from 90 to 135V.

IC_2 is a chemistry-independent battery charger. Though designed to drive an external transistor, in this case, it drives the MOC211 optoisolator to control IC_1 across the isolation barrier. IC_2 s internal circuitry sets the battery voltage to 8.4V–0.5%. The 0.1Ω current-sense resistor, R_1, and I_{SET} resistor, R_2, set the battery current to 1A–2%. (DI #2161)

To Vote For This Design, Circle No. 344

FIGURE 1



A high-voltage controller, IC_1, transformer, and chemistry-independent battery charger, IC_2, charge two-cell Li-ion batteries directly from the ac line.

Emissions killers trap common-mode currents

GLEN CHENIER, FUJITSU NETWORK COMMUNICATIONS, RICHARDSON, TX

An unshielded twisted-pair cable that is transformer-coupled to a digital system can easily act as a radiating antenna, not because of the differential analog signal the cable carries, but because of common-mode currents induced by unwanted stray coupling from the digital portions of the system. These currents from fast digital transitions contain harmonics in the hundreds of megahertz and can be a nightmare to design engineers who have to make systems conform to radiated-emissions limits.

If the coupling transformer has a center tap on the winding to which the cable attaches, you can use this tap to reduce the level of these nasty common-mode currents on the cable. Connecting the tap to a quiet earth ground provides a path to shunt these currents harmlessly to earth before they can sneak out the cable and radiate (Figure 1). A capacitor in the connection provides the same RF grounding function but presents a high impedance to any 60-Hz ground loop currents if the far end of the cable also connects to a ground-referenced transformer winding. This capacitor should be only a few hundred picofarads, must have short leads and circuit traces between transformer tap and good earth ground, and must have a sufficiently high voltage rating to withstand high-voltage transients as the end market requires.

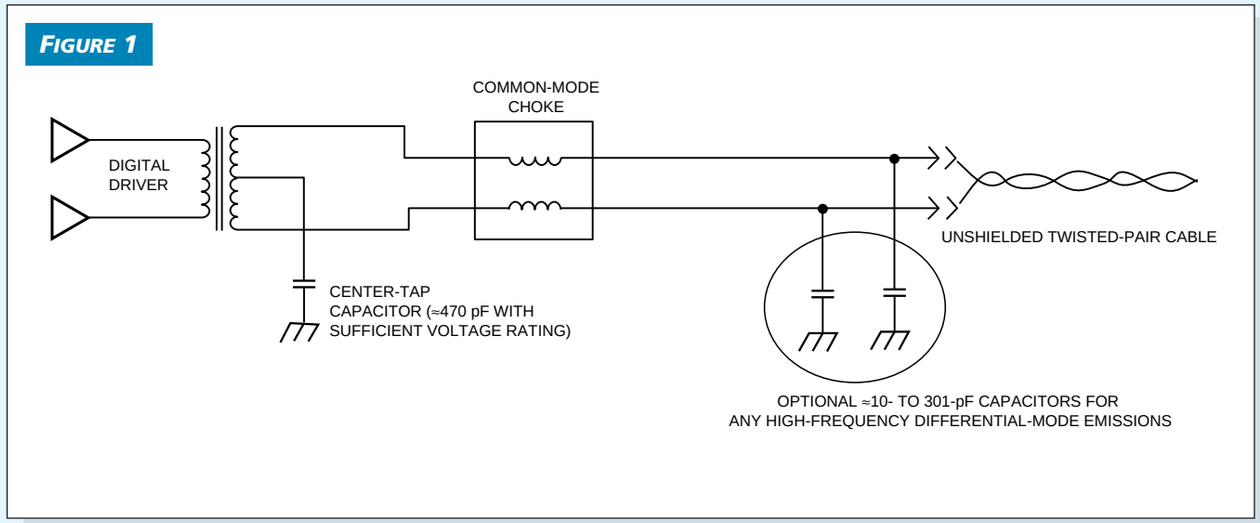
The technique works as follows: The opposite ends of the transformer winding are balanced with respect to ground; that is, the windings push and pull with equal amplitude but opposite polarity on each and every transmitted data symbol. The center of the transformer is the pivot on which the winding balances. As such, this pivot point is neutral relative to ground; an actual connection to ground makes no difference to the differential information signal.

If a common-mode signal impresses both conductors, the resulting currents at opposite ends of the winding flow both toward and away from the center tap in the same phase. This flow causes magnetic cancellation between the two halves of the winding, and the resulting inductance is very low, resulting only in the residual leakage inductance. In this way, both conductors have a low-impedance path to earth ground without affecting the wanted differential signal. Note that filtering each conductor with an RC network also provides the low-impedance path to ground; unfortunately, this filter also destroys the differential signal in high-bit-rate applications.

The technique in Figure 1 also helps reduce susceptibility to common-mode currents that external fields induce; the unwanted currents pass harmlessly through each half of the transformer winding and cancel each other out. Interwinding capacitance the usual mechanism by which common-mode voltages can affect transformer-coupled receiver inputs is less critical because both conductors have a low-impedance path to ground, resulting in minimum common-mode voltage on each conductor.

Using a common-mode choke in addition to the center-tap trap results in a real common-mode killer. The two techniques complement each other, and it can be helpful to use both together in stubborn cases. As Figure 1 indicates, you can place the common-mode choke virtually a transformer on its side in line with the cable, preferably at a point just before the cable exits the (ideally) shielded enclosure to avoid stray-noise pickup on the cable after the choke. A similar but opposite mag-

FIGURE 1



A center-tap capacitor connected to earth ground implements a common-mode current trap and reduces RF emissions. A common-mode choke in addition to this center-tap trap results in a common-mode killer.

netic magic takes place in the common-mode choke, which must present a high series impedance instead of a low shunt impedance to common-mode currents. The winding turns ratio is 1-to-1, and the polarity is such that the magnetic fields from the differential signal now cancel, resulting in almost zero attenuation other than that resulting from the leakage inductance. On the other hand, the common-mode currents cause magnetic addition, which results in high impedance and reduces the level of unwanted currents.

You can also make a common-mode choke by slipping a large ferrite sleeve over the two conductors of the twisted pairs or by winding one or more turns of the twisted pairs through a large toroid doughnut. Many ferrite suppliers make these sleeves and toroids just for this purpose. Also, well-balanced common-mode chokes of the more conventional transformer-like construction are also readily available from datacomm-transformer suppliers.

Two capacitors following the common-mode choke can reduce high-frequency differential-mode emissions caused by non-common-mode currents. (DI #2160)

To Vote For This Design, Circle No. 345

Isolated driver forms solid-state circuit breaker

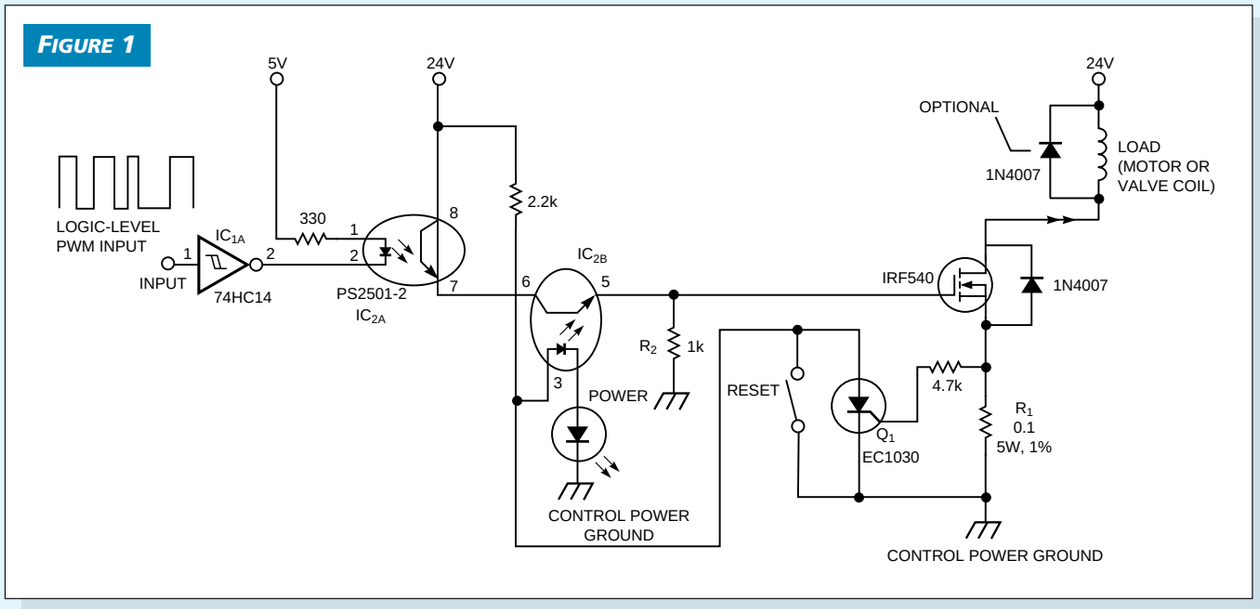
BOB WATSON, CORLEY MANUFACTURING CO, CHATTANOOGA, TN

The circuit in Figure 1 allows standard TTL logic levels to safely drive a high-power dc load. The circuit provides for both signal and ground isolation as well as a solid-state circuit breaker.

The input signal drives IC_{1A}, which in turn provides drive current for optoisolator IC_{2A}. In the absence of an overcurrent condition, IC_{2B} conducts the signal to the gate of the MOSFET. When sufficient current passes through current-sense resistor, R_1, to cause a voltage drop of approximately 0.7V, SCR Q_1 latches on. When Q_1 is on, the circuit pulls Pin 3 of IC_{2B} low, which stops the transistor side of IC_{2B} from conducting. R_2 then holds the gate of the MOSFET low, which prevents it from conducting until you reset the SCR. (DI #2163)

To Vote For This Design, Circle No. 346

FIGURE 1



An overcurrent condition in this isolated PWM driver turns on SCR Q_1, which stops IC_{2B} from conducting.

µC measures high-frequency signals

STAN D'SOUZA, MICROCHIP TECHNOLOGY INC, CHANDLER, AZ

To measure a high-frequency signal using an 8-bit µC, the time period of the measured frequency must be relatively close to the internal clock of the µC. For example, if the internal clock period is 1 µsec, then the maximum frequency that you can measure is 1 MHz in most µC applications.

However, you can use the circuit in Figure 1a to measure a frequency much higher than the internal clock frequency of the µC. This circuit uses an external binary ripple counter operating in asynchronous mode. The circuit gates the incoming frequency to the counter using NAND gates and control lines from the µC. To enable counting, the µC sets $CNTL_1$ and $CNTL_2$ to 1. Once the measurement time is complete, $CNTL_1$ resets to 0, which stops further inputs to the counter.

To read the value of the low 12 bits of the frequency, the µC toggles $CNTL_2$ N times until the internal 8-bit timer increments by one. The low 12-bit value of the frequency is then equal to 4096 N. Reading the value of the counter in this manner requires only one I/O line as opposed to 12 lines to read the counter's 12 bits. In combination with the internal 8-bit counter in the µC, a 20-bit frequency measurement is possible. The limiting factor of the measurement is the maximum frequency input to the NAND gates and the ripple counter.

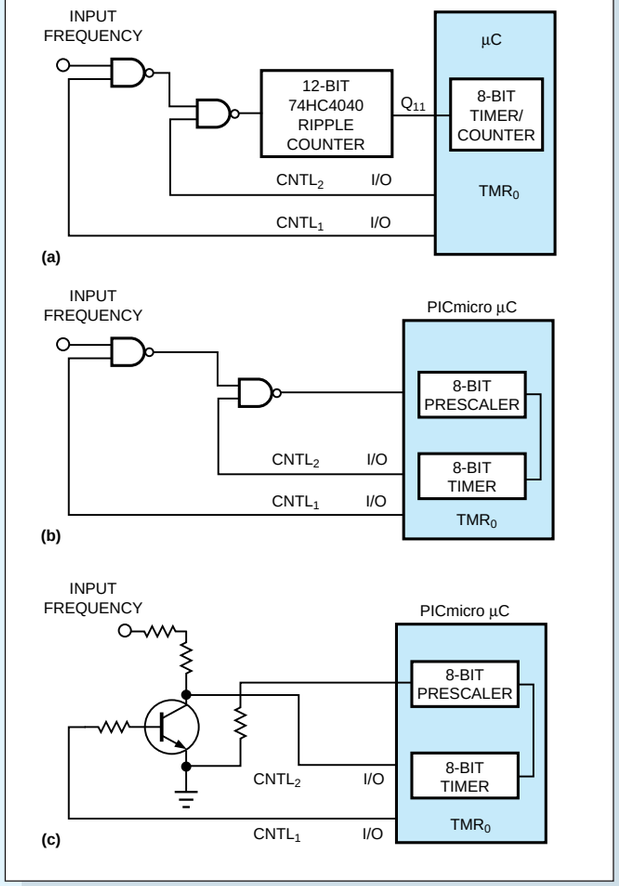
Although the circuit in Figure 1a is straightforward, the circuit does involve the additional cost of a 14-pin NAND gate and a 16-pin counter, which may be undesirable in many applications. In PICmicro 8-bit µCs, the internal 8-bit counter/timer, TMR_0, has an associated 8-bit divider, or prescaler. The counter has read/write capability, but you can't read the prescaler value. You can modify Figure 1a's circuit using a PICmicro µC to implement a 16-bit frequency counter (Figure 1b).

This circuit uses the internal 8-bit prescaler to divide the incoming frequency. The circuit feeds the output of the prescaler to the 8-bit timer, TMR_0, for measurement. As with Figure 1a's circuit, once the gate time is over, $CNTL_1$ blocks additional clocks from the input signal. Then, $CNTL_2$ pulses the 8-bit prescaler N times until the 8-bit timer/counter, TMR_0, increments by 1. In this case, the lower 8-bit value of the measured frequency equals 256 N. The µC then concatenates the value of the counter with the 8-bit timer's value to give a 16-bit value of the measured frequency.

Figure 1c shows a further simplification of the circuit in Figure 1b by replacing the NAND gates with two transistors and four resistors. To start the counter, the µC configures $CNTL_1$ and $CNTL_2$ as inputs or in a high-impedance mode. Thus, the circuit directs the incoming signal to the prescaler and in turn to the 8-bit timer/counter, TMR_0. When the gate time is complete, the µC makes $CNTL_1$ an output going low. A low $CNTL_1$ deactivates the transistor, whose output becomes an open collector. The µC can now make $CNTL_2$ an output normally high and going low to pulse the input to the prescaler. The µC pulses $CNTL_2$ low N times until the value of TMR_0 increments by 1. The low 8-bit value of the frequency is equal to 256 N. To begin counting again, the µC reads the value of TMR_0, which clears the prescaler, and again configures $CNTL_1$ and $CNTL_2$ as inputs.

Note that the architecture of the PICmicro µC allows accurate timekeeping for the gating pulse because the timing of a software loop is predictable and accurate to within one instruction cycle; a PICmicro µC executes each instruction in one cycle, except for branch instructions, which take two cycles. (DI #2164)

FIGURE 1



A binary ripple counter and NAND gate team with a µC (a) to measure input frequencies higher than the µC clock. Using a PICmicro µC reduces the number of necessary components (b). A further simplification (c) replaces the NAND gate with one transistor and four resistors.

To Vote For This Design, Circle No. 347

Microamps monitor dual-supply batteries

BRUCE ANDERSON, UNIVERSITY OF WISCONSIN—MADISON

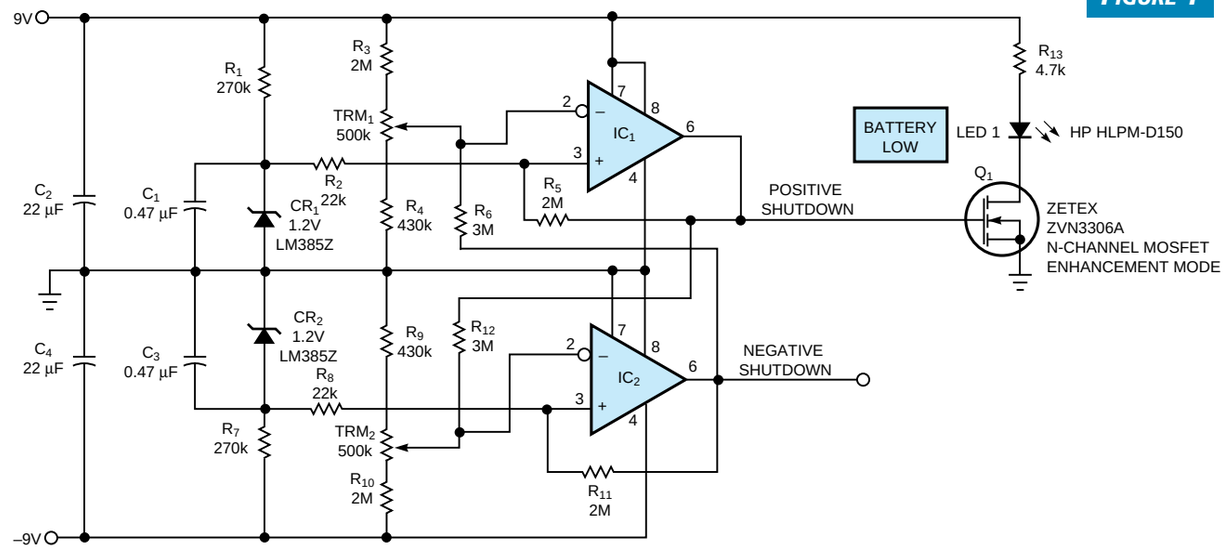
The low-power circuit in Figure 1 monitors two 9V batteries in a dual-supply configuration and turns on the Battery Low LED if either battery voltage drops below its limit. It also provides two shutdown signals you can use to turn off voltage regulators, such as Maxim s MAX663/664 positive and negative regulators. By using low-power voltage references and op amps, the circuit holds the current drain to approximately 45 µA from each battery, with the positive drain rising to approximately 1 mA when the LED turns on.

Each battery voltage undergoes comparison with a Motorola LM385Z 1.2V reference, using a Maxim 7612 op amp with hysteresis via a 2-MΩ resistor (R_5 and R_{11}). Cross-coupling via the 3-MΩ resistors (R_6 and R_{12}) ensures that if either shutdown signal goes true, both do, and the circuit locks up in the shutdown state with the Battery Low LED illuminated. C_1 and C_3 delay the reference voltages so that when the batteries switch on, the circuit comes up in the proper state. Positive Shutdown is at the positive rail when true. Negative Shutdown is at the negative rail when true. The values shown allow you to adjust the battery-low limits over a range of approximately 3.8 to 8.1V. For our applications with Eveready EN22 alkaline batteries, we typically set the limits at 6.5V. This setting uses a good portion of the battery life, yet allows some reserve for continued operation after the LED comes on.

The circuit has two convenient features that were unforeseen before testing. One is that when the batteries switch off the LED flashes briefly as the decoupling capacitors discharge. The flashing indicates that the batteries are not so totally dead that they cannot light the LED. The other is that the LM385 has an initial turn-on voltage about 10% higher than the steady-state 1.2V reference. Thus, when you switch the batteries on, they must have a voltage about 10% higher than the steady-state threshold to be considered good. So if the Battery Low LED stays off when the device turns on, the batteries will remain good for a while. Of course, if you are not using the shutdown signals to turn off regulators, you can set the threshold so that your device will continue to operate for a period after the LED comes on. Using the battery-discharge characteristics and your circuit s voltage and current requirements, you can select a threshold that gives appropriate reserve for your application. (DI #2169)

To Vote For This Design, Circle No. 348

FIGURE 1



NOTES:

1. IC_1 AND IC_2 ARE MAXIM ICL7612ECPA.
2. CR_1 AND CR_2 ARE MOTOROLA LM385Z-1.2.
3. C_1 AND C_3 ARE 0.47 µF, 50V CERAMIC.
4. C_2 AND C_4 ARE 22 µF, 16V TANTALUM.
5. ALL RESISTORS ARE 1/4W, 5 OR 10% CARBON COMPOSITION.
6. TRM_1 AND TRM_2 ARE BOURNS 3296W.

If either battery voltage in a two-battery supply drops below a preset limit, a Battery Low LED turns on.

Program provides integer-to-binary conversion

BERT ERICKSON, FAYETTEVILLE, NY

Binary numbers rarely appear in applications of C or C++ programs, so any reference to converting from an integer to a binary number is usually relegated to a few simple examples in the appendix. However, when you re working with codes for communication systems, terms such as parity, checksum, distance, weight, and block codes are much easier to verify with a check solution when they are in binary form. C and C++ statements do use integers for manipulations that have binary implications. However, when the analysis gets down to the binary-number level, the conversion from integers is hard to find in the libraries supplied with the compiler. The cintbin and classicC functions in Listings 1 and 2 convert an integer in the main function to a binary number that remains available in the main function.

The ones and zeros in the elements of the array correspond to the location of bits in the customary binary number. You can compile the C++ cintbin version as listed. Readers who have an ANSI C compiler can use the program preceded by // in Listing 2. For long integers, refer to the revised edition of Microsoft C Programming for the PC by Robert Lafore. The first part of the listing is only a driver that has a call to the function and a printout for the binary number. You can use the bits in the binary number in any additional statements.

The first argument in the call should be 31 or less to provide some leading zeros but large enough to make sure the most significant bit is included. The temporary variable z and the return value provide some assurance that the result is valid. The statements in both functions are self-explanatory, so the only thing left to do is to compile one of the programs and enter 31 4,294,967,295 with a space after 31 to verify the 32-bit binary number 1111 1111 1111 1111 1111 1111 1111 1111. You can download the listings and the executable cintbin file from EDN s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the files from DI-SIG, #2156. (DI #2156)

To Vote For This Design, Circle No. 349

LISTING 1—C++ INTEGER-TO-BINARY CONVERSION ROUTINE

```
// cintbin.cpp            a C++ function
// Converts an integer to a binary number
#include <iostream.h>
#include <math.h>
 int c[32];
main() {
 int n; unsigned long int x;
 unsigned long int cintbin(int, unsigned long int);   //Declare

 cout << "\n\tEnter max power of 2 desired (31 or less) and "
      << "the integer number\n" << "\t(4,294,967,295 or less) "
      << "with a space between them  ";
 cin >> n >> x;
 cout << "\tReturn x = " << cintbin(n, x)
      << " should equal the input value shown above\n   Bin # =";
 for (int k = n; k>= 0; k--) cout << ' ' << c[k];
  return 0;
 }

 unsigned long int cintbin(int n, unsigned long int x) //Define
     {
     unsigned long int y, z; z = x;
     for (int i = 0; i < n+1; i++) c[i] = 0;

     for (int j = n; j >= 0; j--)
  { y = int(pow(2,j)); if (x >= y)
    { c[j] = 1; x = x - y; }
  }
 return z;
```

LISTING 2—CLASSICC INTEGER-TO-BINARY-CONVERSION ROUTINE

```
//// classicC.cpp   a C function
//// Converts an integer to a binary number
//#include <stdio.h>
//#include <math.h>
// int c[32];
//main()  {
// int n; unsigned long int x;
// unsigned long int classicC(int, unsigned long int);   //Declare
//
// printf("\n\n\tEnter max power of 2 desired (31 or less) and ");
// printf("the integer number\n");
// printf("\t(4,294,967,295 or less) with a space between them  ");
// scanf("%d %lu",&n,&x);
// printf("\tReturn x = %lu",classicC(n,x));
// printf(" should equal the input value shown above\n   Bin # =");
// for (int k = n; k >= 0; k--)
//  printf(" %d",c[k]);
// return 0;
// }
//
// unsigned long classicC(int n, unsigned long x) //Define classicC
// {
//  unsigned long y, z;
//  int i, j;
//  z = x;
//  for (i = 0; i<n+1; i++)
//    c[i] = 0;
//  for (j=n; j>=0; j--)
//   {
//    y=pow(2,j);
//    if (x >= y)
//    {
//     c[j] = 1;
//     x = x - y;
//    }
//   }
//  return z;
 }
```

Inexpensive relays form digital potentiometer

ROBERT PERRIN, Z-WORLD, DAVIS, CA

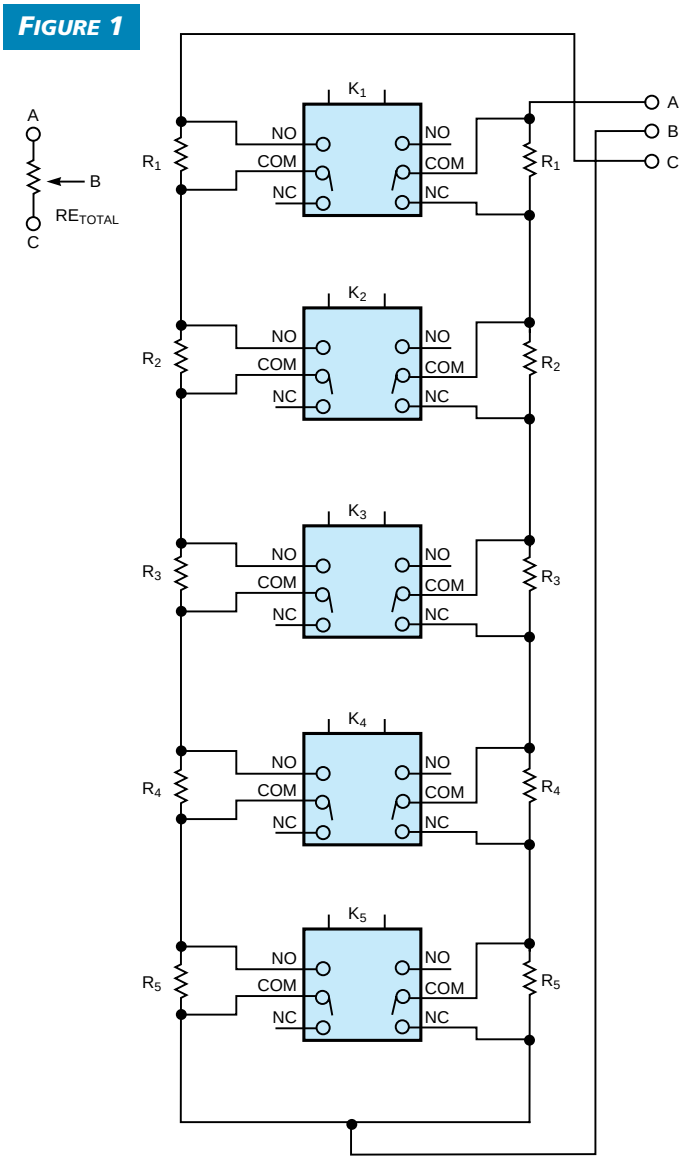
A project posed the challenge of replacing existing analog potentiometers (used to set brightness and contrast levels) in video monitors with digitally controlled potentiometers. The different brands and models of monitors presented widely varying voltages across the potentiometers. The design had to

- accept digital controls,
- be purely resistive,
- tolerate 60V dc or ac between any two points,
- be able to dissipate 1W,
- be scalable for future systems, and
- have high isolation between the digital controller and the simulated potentiometer.

Also, during switching, the wiper could not become open-circuited, and it had to be able to travel end-to-end.

Existing digitally controlled potentiometers were not up to the task. FET switches faced special challenges with the ac-voltage isolation criterion. The design in **Figure 1** uses inexpensive relays to build a digitally controlled potentiometer. The relays switch their respective resistors above or below the tap. With all the relays de-energized, the potentiometer has the

FIGURE 1

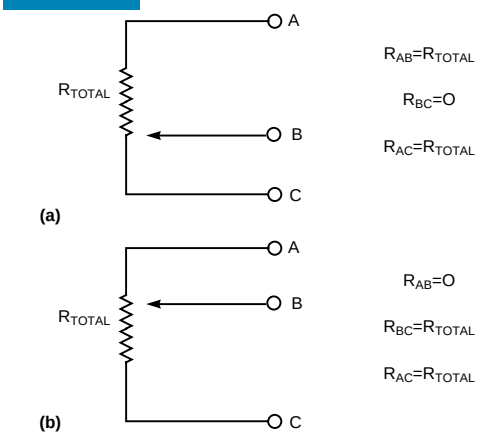


A handful of relays and fixed resistors replaces an analog potentiometer. The resolution of the simulated potentiometer is one part in 2^n, where n is the number of relays.

TABLE 1—RESISTOR VALUES FOR DIGITAL POTENTIOMETER

Reference designator	Resistance
R_1	$R_{TOTAL}/2^1$
R_2	$R_{TOTAL}/2^2$
R_3	$R_{TOTAL}/2^3$
R_4	$R_{TOTAL}/2^4$
R_5	$R_{TOTAL}/2^5$
R_n	$R_{TOTAL}/2^n$

FIGURE 2



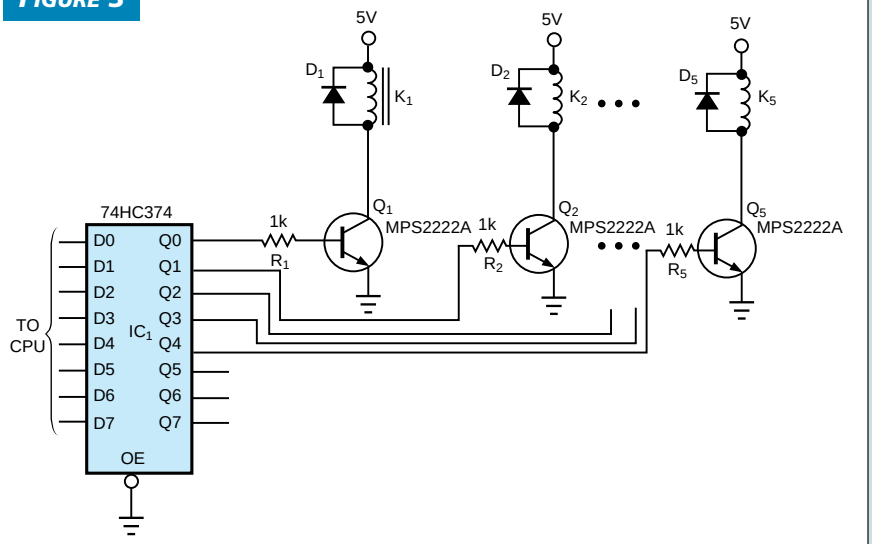
De-energizing all the relays in Figure 1's circuit results in the configuration in a; energizing all the relays results in the configuration in b.

configuration shown in **Figure 2a**. With all the relays energized, the relay takes the form shown in **Figure 2b**. The resistors are weighted in a $R_{TOTAL}/2^n$ relationship. This weighting gives a good approximation of a linear taper with 32 (2^5) positions. **Table 1** shows the selection of resistor values.

The circuit uses a 74HC374 latch and MPS2222A npn transistors to interface the relays to the system CPU (**Figure 3**). The primary disadvantages of the method are board space and cost. However, for the monitor application, the circuit proved flexible and reliable. Applications that require replacing existing potentiometers or rheostats are potential candidates for this circuit. For example, you could replace a high-power rheostat with the circuit if you select relays and resistors with suitable current ratings. (DI #2167)

To Vote For This Design, Circle No. 406

FIGURE 3



A TTL latch and a collection of inexpensive npn transistors, resistors, and relays allow you to configure a simulated potentiometer with any desired degree of resolution.

Switched-capacitor regulator provides gain

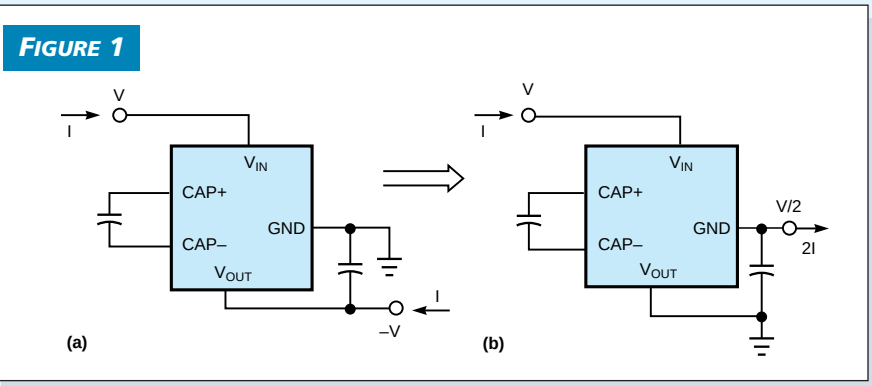
JEFF WITT, LINEAR TECHNOLOGY CORP, MILPITAS, CA

Linear voltage regulators become inefficient when the input voltage is much higher than the regulated output. The circuit in **Figure 1** dramatically increases efficiency when the input voltage is more than twice the desired output; for example, using 12V to obtain a regulated 3.3 or 5V. You usually use a switched-capacitor voltage inverter to generate a negative supply voltage from a positive input voltage. The negative-supply current is equal to the current drawn from the input. By swapping the roles of the ground and output pins, the inverter in **Figure 1** divides the input voltage by two. It also doubles the current from the input to the output, thereby providing much better efficiency than does a linear regulator.

Figure 2 illustrates the circuit's operation. An internal oscillator alternately closes and opens four switches. In the first half-cycle, switches 1 and 2 close, and current flows from the input to the output, charging C_1. In the second half-cycle, switches 3 and 4 close, discharging C_1 into the output. The current delivered to the output is continuous and equal to twice the average input current. Because the current is continuous, the output-voltage ripple is low. Note that you do not need to match C_1 and C_{OUT}, because their voltages equalize on each cycle.

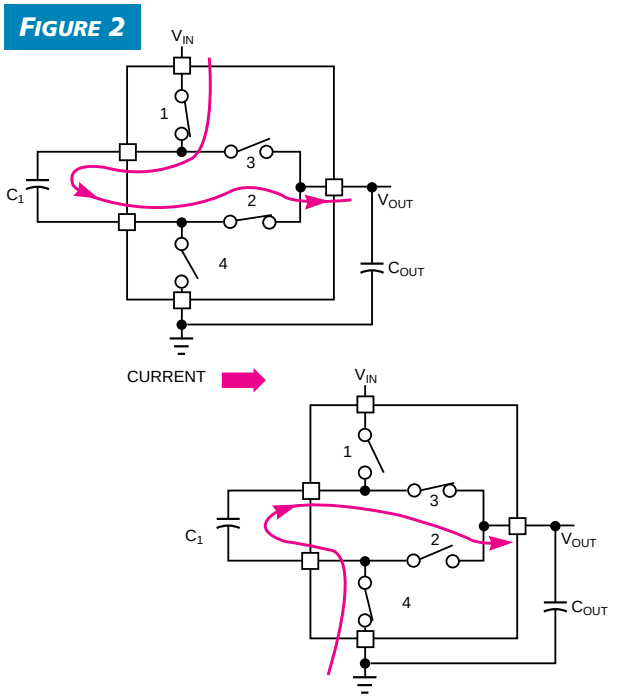
Figure 3 shows the actual circuit. IC_1, an LT1054 switched-capacitor voltage converter and regulator, modulates the

FIGURE 1



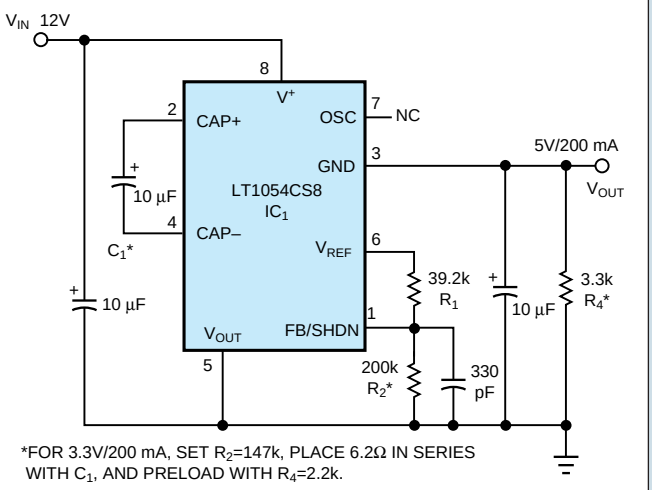
Rewiring a switched-capacitor inverter for step-down regulation results in a current gain of 2.

FIGURE 2



This switched-capacitor regulator doubles the current from the input to the output, thus increasing efficiency and eliminating the need for a heat sink.

FIGURE 3



The LT1054's internal switches alternately charge and discharge C_1, thereby delivering a continuous current to the output.

current (through switch 1 of **Figure 2**) to regulate the output. A servo loop keeps the potential at the FB pin equal to the potential at the GND pin. The circuit can deliver 200 mA at 5V from an input of 11.2 to 13V. Typical efficiency is 74%, compared with 42% for a linear regulator. More important, dissipation decreases from 1.4W for a linear regulator to 0.35W, a figure that IC_1's eight-pin, surface-mount package can easily handle. For a 200-mA, 3.3V output, the circuit is 49% efficient, compared with a linear regulator's 27%, with power dissipation reduced from 1.8 to 0.7W. A 6.2Ω resistor in series with C_1 shares the dissipated power with the LT1054; the circuit needs no heat sink. (DI #2168) ▤

To Vote For This Design, Circle No. 407

Precision current sink costs less than $20

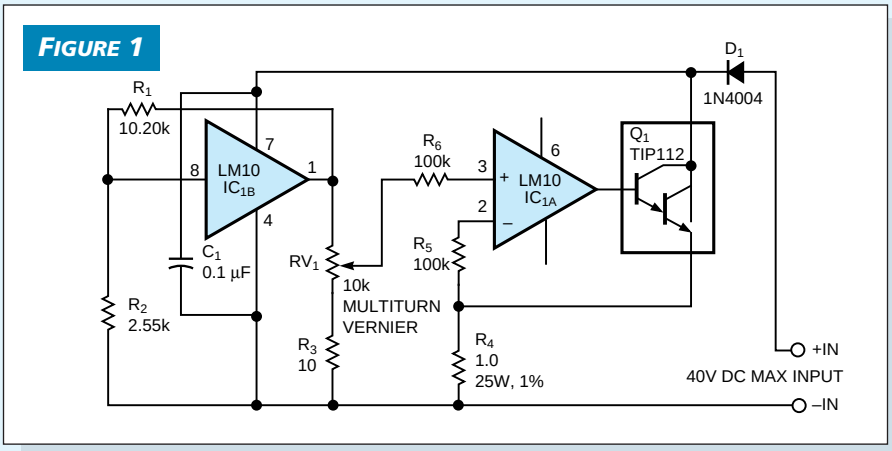
CARLOS BARBERIS, BARTEK TECHNOLOGIES, HAVERHILL, MA

If you often need a simple active load (constant-current sink), you can benefit from the simple circuit in **Figure 1**. The need often arises to measure the life of a battery or other power device under constant-load conditions. The easy-to-build and inexpensive circuit in **Figure 1** is a handy addition to your arsenal of test fixtures. You can build the circuit for less than $20. The most expensive parts are the vernier knob and the multiturn potentiometer. You can build the active load into a miniature enclosure with banana-jack connectors. The vernier control allows you to directly set current from 1 mA to 1A by simply dialing the desired set current. Without the vernier and multiturn potentiometer, you could build the circuit for less than $10, but you then lose the advantage of a calibrated, stand-alone test box.

The circuit is a precision current sink with typical current regulation of better than 0.5% for a 3 to 40V compliance voltage. R_4 is a sensing resistor; its voltage drop servos the input voltage to IC_{1A}. The wiper of the vernier potentiometer sets the input voltage, discounting any amplifier offset errors. The offset could be as high as 2 mV in a run-of-the-mill LM10, translating to a 2-mA error between the set current and the current flowing in R_4. The reference amplifier, IC_{1B}, is a gain-of-5 stage that provides a 1.00V reference on the high side of the current-setting potentiometer. The voltage-to-current transfer function is thus 1A/1V. You can change the transfer function to fit your needs.

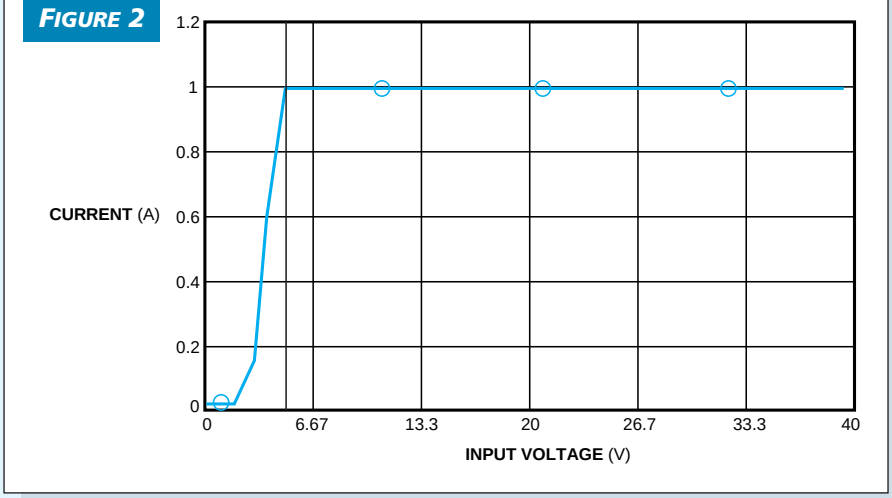
Although the current-control mechanism allows the output of the current source to approach zero, the additional currents consumed by the circuit (approximately 400 µA) establish the baseline current. Therefore, you set the bottom of the potentiometer via R_3 to start at approximately 1 mA. Under normal operation, a current setting of 1 to 300 mA maintains the setpoint within 0.5% with 3 to 40V compliance. Currents above 300 mA require 3 to 5V for compliance. The circuit maintains a 1A current within 300 µA from 4.9 to 40V or within 0.001% tolerance (**Figure 2**). You could lower the initial regulation point by one diode junction by removing D_1, whose sole purpose is to prevent destruction of the active circuitry when you connect the power supply backward.

FIGURE 1



A handful of inexpensive parts builds a precision current sink that provides 1-mA to 1A sink current over a wide compliance-voltage range.

FIGURE 2



For compliance voltages above 4.9V, the circuit in Figure 1 provides a rock-solid 1A sink current, with less than 0.001% variation with voltage.

The principal sources of error in the circuit are the amplifier offset, the tolerance of the reference voltage, the tolerance of R_4, and the fact that the current includes various branch currents other than the controlled current in the sensing resistor. These branch currents add up to approximately 400 µA, or roughly five times lower than the offset-voltage error. You can consider the error negligible for settings of 10 mA and above. The most important issue for long-term stability is efficient heat removal from the current-regulating transistor, Q_1. The transistor needs an appropriate heat sink; the choice of heat sink depends on the current ranges you need.

The element that encounters the largest voltage drop at a given current is the hottest. Q_1 dissipates V_{IN}–1W for any given input voltage when operating at 1A. If you plan to use the load on a continuous basis, for example at 1A, with a 30V input, Q_1 dissipates 29W; R_4 consumes 1W. Q_1 would thus need a hefty extruded heat sink. (DI #2171)

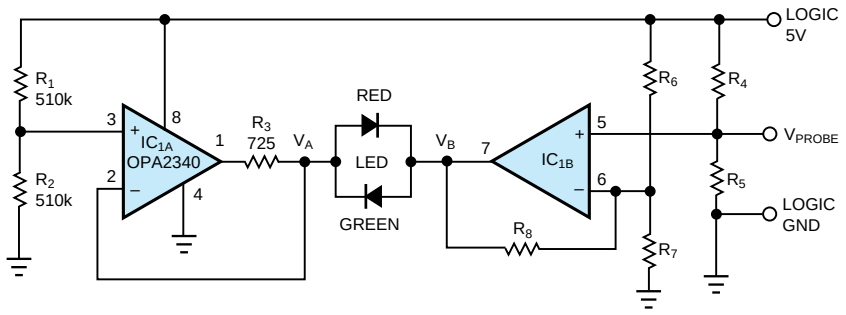
To Vote For This Design, Circle No. 408

Simple logic probe uses bicolor LED

MARK SHILL, BURR-BROWN CORP, TUCSON, AZ

When probing digital logic levels on a circuit board, locating the indicator of the logic level near the probe tip is convenient, so that you can keep both the indicator and probe tip constantly in sight. When using a handheld DVM or oscilloscope probe, you must momentarily look away to read the logic level. In that instant, the probe can slip and cause a short circuit. The circuit in **Figure 1a** implements a simple handheld logic probe using just a few components. This

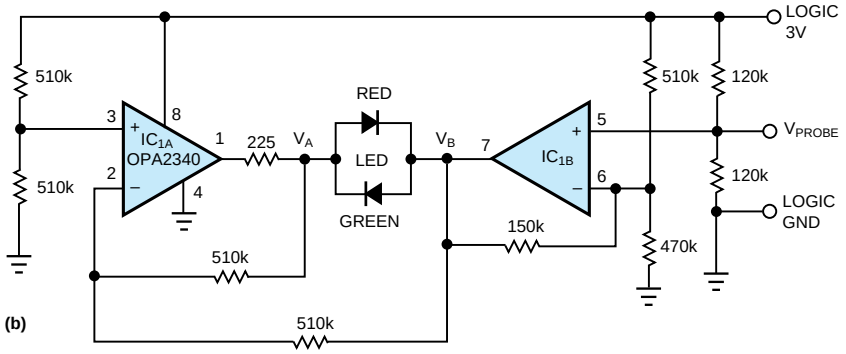
FIGURE 1



NOTE:
LED=RADIO SHACK 276-012.

RESISTOR	5V CMOS	TTL
R_4	160k	250k
R_5	120k	100k
R_6	510k	820k
R_7	270k	220k
R_8	180k	470k

(a)



(b)

A simple handheld logic probe uses a rail-to-rail op amp and a bicolor LED (a). A modified probe circuit works with 3V CMOS logic (b).

probe can measure high, low, and high-impedance logic states, in addition to indicating switching logic states. This probe is useful for quick measurements of dc logic levels. You can use a similar probe circuit (**Figure 1b**) for 3V CMOS logic.

The circuit centers around the OPA2340 dual rail-to-rail op amp and a Radio Shack 276-012 bicolor LED. The forward voltage for the red and green LEDs are 2 and 2.1V, respectively. Op amp IC_{1A} derives a buffered 2.5V reference output from the 5V supply, and R_3 limits the current to the LED when it is on. IC_{1B} buffers and amplifies the probed logic signal to a 0 to 5V output level. R_4 and R_5 set a reference level for the positive input of IC_{1B} for the case of a high-impedance level. When a logic high is present, the green LED lights; a logic low lights the red LED. When a high-impedance state is present, the LED is off.

The output voltage transfer function of IC_{1B} is

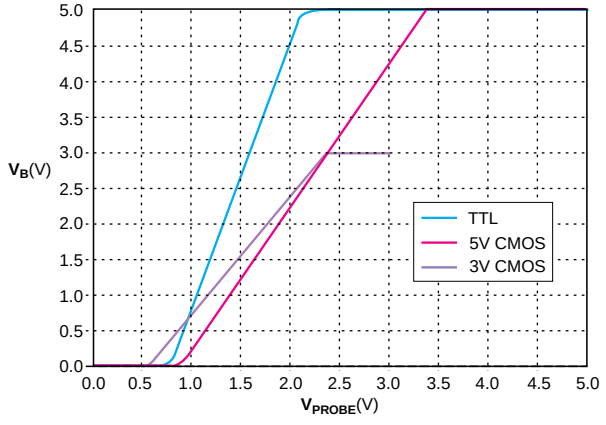
$$V_B = \left[1 + \frac{R_8(R_6 + R_7)}{R_6 \bullet R_7}\right] V_{PROBE} - \frac{R_8}{R_6} \bullet V_{CC},$$

which makes $V_{LED}=V_B-2.5V$. The values of R_6, R_7, and R_8 are such that V_B limits at the positive or negative power-supply rail when V_{PROBE} is at the logic family's minimum low or high level, respectively. **Figure 2** shows the voltage-transfer function for V_B. When V_B limits at the rail voltage, the LED lights red or green, depending on the probed logic level. When V_B is within 0.5V of the negative rail, the red LED turns on; when V_2 is within 0.4V of the positive rail, the green LED turns on. If the probe measures a high-impedance state, voltage-divider resistors R_4 and R_5 set the positive input of IC_{1B}, and the voltage across the LED is approximately 0V.

In **Figure 1b**, the output of IC_{1B} feeds into IC_{1A}'s inverting input. For the resistor values in **Figure 1b**, the transfer function for IC_{1A} is $V_A=-V_B+3V$, thus making $V_{LED}=2V_B-3V$. (DI #2162)

To Vote For This Design, Circle No. 409

FIGURE 2



The output of IC_{1B}, V_B, limits at either the positive or negative supply rail depending on the probe voltage.

DC power wire also carries clock or data

MIKE HARDWICK, DECADE ENGINEERING, TURNER, OR

A high-side current-sense amplifier, IC_1, offers a simple method of combining low-speed clocks or other signals with dc power in cables between subsystems (**Figure 1**). Designed for monitoring charge and discharge current in secondary batteries, IC_1 outputs a current of 0.5 mA per amp of load current flowing through its internal sense resistor while rejecting common-mode supply-voltage noise. The on-chip sense resistor handles as much as 3A of continuous current. The IC accommodates power-supply voltages from 3 to 36V.

Figure 1 depicts a subsystem that receives power from its host system and simultaneously transmits a clock signal back on the same wire. The circuit uses the clock in the remote system to modulate power-supply current via an open-collector driver or discrete transistor and R_{MOD}, a switched load resistance. In the host system, IC_1 develops a voltage across R_{IV}, which represents the instantaneous sum of supply current and modulation current. R_{INTEG} and C_{INTEG} filter this voltage, biasing the comparator's reference pin to a level that tracks average power-supply current. As the signal swings above and below this reference level, the comparator outputs the recovered clock. R_{HYST} adds a small amount of hysteresis to ensure clean clock recovery.

IC_2, which comes in an SOT23-5 package, is a CMOS comparator with a rail-to-rail input range. This range allows you to choose R_{IV} with relative freedom—expect about 1V/A of load current for each 2 kΩ of resistance. The input offset of low-grade versions of IC_2, an LMC7211, can be as much as ±18 mV. Thus, select R_{MOD} and R_{IV} to produce 50 mV or more of modulation on R_{IV}, and then choose R_{HYST} to obtain a few additional millivolts of shift at this node when the comparator changes state. None of these values is critical.

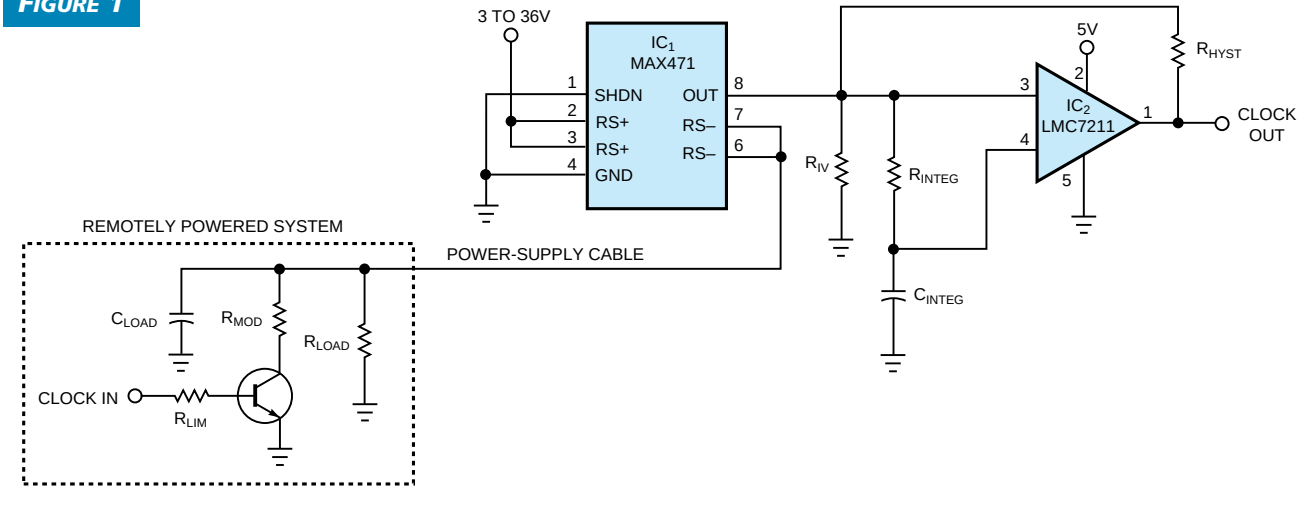
For this scheme to work as expected, the power-supply current to the remote system must be relatively constant, except for the intentional modulation. Slow power-supply current variations do not cause problems, as long as you choose the integrator components with care. The integrator RC product should be about 10 times the clock period. It's convenient to choose R_{INTEG} of approximately 1 MΩ when using CMOS comparators, such as IC_2. You can then use ceramic or plastic-film capacitors for C_{INTEG}, thus minimizing the risk of failure due to capacitor leakage.

The circuit can also transmit data if the data contains little dc bias variation or if you replace the integrator components with a fixed bias source. This change means, of course, that no significant power-supply current change is allowable after calibration.

IC_1's output rise and fall times measure approximately 4 µsec, a figure similar to IC_2's response-time specification. The remote system's power-supply bypass capacitance may impose an upper bound on the clock rate because this capacitance limits the modulation rate of the supply current. This time constant is $C_{LOAD} \times (R_{SENSE}+R_{CABLE})$. R_{SENSE} is less than 0.07Ω in IC_1, and the cable's series resistance depends on the application. You must add the equivalent series resistance of the power source, connectors, and any other associated resistive elements to R_{SENSE} in this calculation if you're pushing the envelope of the circuit's performance. (DI #2175) 目

To Vote For This Design, Circle No. 410

FIGURE 1



By using a high-side current-sense amplifier IC (IC_1) in an unconventional manner, you can combine clock or data signals with dc power in cables.

PLL implements FPGA-based SDRAM controller

EDDY DEBAERE, BARCO GRAPHICS, GHENT, BELGIUM

As FPGA capabilities increase and time to market decreases, FPGAs gain more acceptance for implementing both data and control paths. Thus, they find wide use as controllers and datapath glue logic for fast-page DRAMs. Synchronous DRAMs (SDRAMs), whose control signals use a clock input as reference, are a natural target for FPGA-based controllers. SDRAMs operate at frequencies of 100 MHz and higher (in contrast with fast-page DRAMs, for which a 60-MHz memory-system clock was considered high). **Figure 1** shows a way to implement FPGA-based SDRAM controllers. **Figure 2** shows the timing for a Xilinx XC4010E-2 device. You can apply the method to FPGAs from other vendors, as well as to high-frequency systems other than SDRAMs.

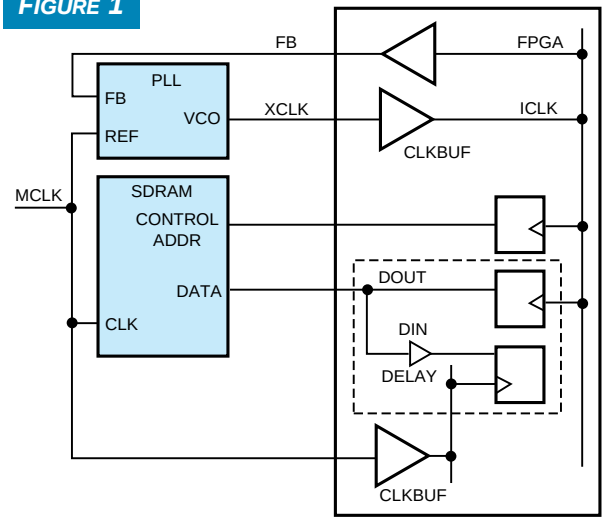
When you use FPGAs, delays to get on and off chip add up quickly: clock pin to internal clock buffer for the internal clock-distribution net (5.4 nsec) and internal clock to output flip-flop (4.5 nsec minimum). The sum of these delays excludes an FPGA from application with SDRAMs using a 10-nsec clock period, considering a 3-nsec setup time for the address, control, and data-out signals. **Figure 1** shows how to use a PLL with an FPGA and how to implement clock and control-path signals to make FPGA-based SDRAM controllers that operate at 100 MHz and beyond. The SDRAM address, data, and control signals use the memory-system clock, MCLK, as reference and the FB signal from an FPGA output pin as feedback.

Because the PLL loop uses no dividers, MCLK and XCLK have the same frequency. When the PLL locks, the XCLK signal precedes MCLK by a time equal to the sum of the clock-buffer delay (from XCLK to ICLK) and the combinatorial output-buffer delay (from ICLK to the PLL feedback pin). The address, control, and data-out signals obtain their clock signals from ICLK. Because the delay from ICLK to the clocked output pin (7.0 nsec for the slew-rate-limited output) is close to but higher than the delay from ICLK to the FB output pin (4.8 nsec for a fast output), the clocked output appears early in the CLK period.

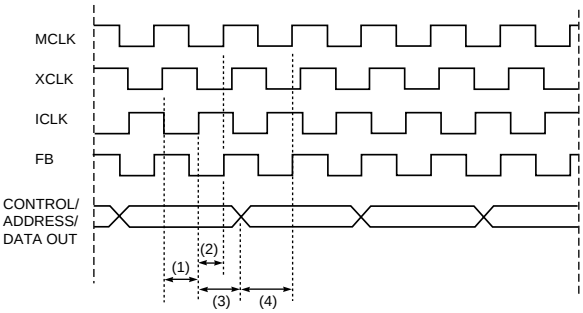
The required SDRA setup time is thus easily fulfilled in a 100-MHz system: (10–(7.0–4.8))>3 nsec. Note that changes in temperature, IC processes, and voltage have little influence on performance, because the XCLK clock-generating circuit is a closed-loop system. Moreover, the output buffer of the FB pin and the control and data pins have the same clock input as does the input pin, and all circuit blocks reside on the same die. Tests designed to check the data-input setup-and-hold times used a clocked input flip-flop (in the same I/O block as the data-output flip-flop), which received its clock from an internal version of MCLK. This configuration avoids excessive hold-time requirements for the SDRAM. *(Editor's note: An* EDN *contributing editor warns that the 7.0–4.8 nsec reflects* maximum *specified times for the XC4010E-2 and yields an acceptable margin for an SDRAM with 1.5-nsec minimum hold time. However, an FPGA running at* typical *specs may present a marginal situation.)* Tests using the AV9170-01 PLL from ICS show satisfactory performance to 106.25 MHz. (DI#2165) EDN

To Vote For This Design, Circle No. 415

FIGURE 1



The use of a PLL with an FPGA eliminates the setup-time problems inherent in an FPGA-only configuration, allowing you to design SDRAM controllers for clock frequencies of 100 MHz and beyond.



(1) XCLK TO ICLK INTERNAL CLOCK-BUFFER DELAY: 5.4 nSEC.
(2) ICLK TO FB COMBINATORIAL FAST OUTPUT-BUFFER DELAY: 4.8 nSEC.
(3) ICLK TO SLEW-RATE-LIMITED OUTPUT DELAY: 7.0 nSEC.
(4) ACHIEVED CONTROL/ADDRESS/DATA-OUT SETUP TIME TO SDRAM: MCLK PERIOD MIMUS 2.2 nSEC.

FIGURE 2

The timing for a Xilinx XC4010E-2 device shows 7.0–4.8=2.2 nsec, an acceptable figure for an SDRAM with 1.5-nsec minimum hold time.

Comparator uses signal-dependent hysteresis

P KREHLIK AND L SLIWCZYNSKI, UNIVERSITY OF MINING AND METALLURGY, KRAKOW, POLAND

Sometimes, you need to distinguish between two voltages, using some hysteresis in the decision. When the levels of the compared signals vary over a wide range (for example, a few orders of magnitude), the hysteresis width should vary similarly to ensure a constant ratio between hysteresis width and signal level. You could encounter such a situation, for example, when you need to decide which of two transmission channels conveys a "higher quality" signal (one with a higher level). You need the hysteresis to avoid permanent changes in the decision when the signal levels are close to each other. In this case, the best way to choose the greater voltage is to make the decision by taking the ratio of the signals instead of comparing them directly. **Figure 1** shows a circuit with signal-variant hysteresis.

Amplifiers IC_{1A} and IC_{1B} with associated resistors and diodes, R_1, R_2, D_1, and D_2, operate as logarithmic converters, producing output voltages

$$V_{O1,\,O2} = -V_T \ln\left[\frac{V_{I1,\,I2}}{RI_S}\right],$$

where V_T=kT/q~25 mV at room temperature, and I_S is the saturation current of the p-n junction. The derivation of the equation assumes that R_1=R_2=R. Next, voltages V_{O1} and V_{O2} combine in the summing amplifier, IC_{1C}. Using the assumption that R_3=R_5 and R_4=R_6 and that the diodes are matched,

$$V_{O3} = \frac{R_4}{R_3}(V_{O2} - V_{O3}) = V_T \frac{R_4}{R_3} \ln\left[\frac{V_{I1}}{V_{I2}}\right].$$

the output voltage from IC_{1C} is

The voltage is thus proportional to the ratio of the input voltages. The last amplifier in the circuit, IC_{1D}, is a standard

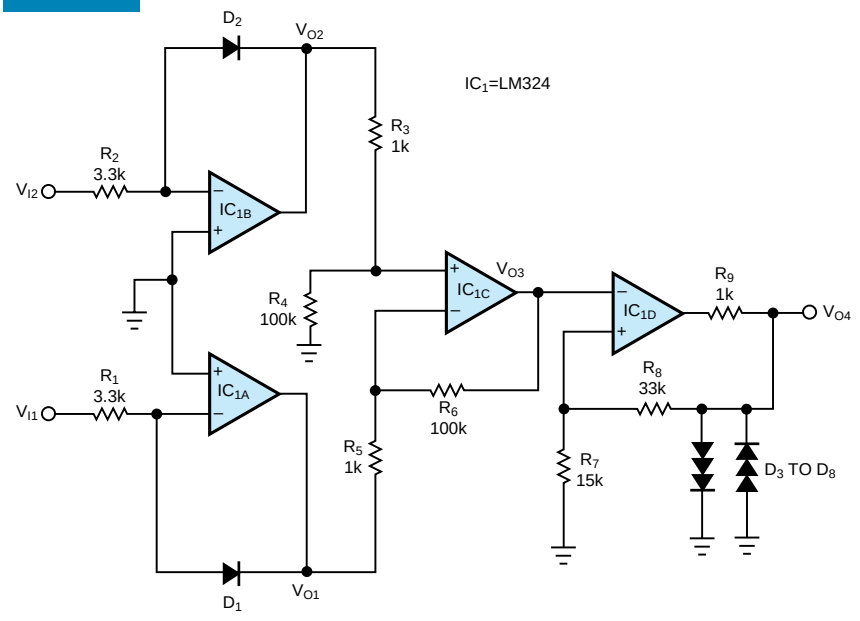
$$V_H^+ = V_{O4}^+ \frac{R_7}{R_7 + R_8}, \text{ and } V_H^- = V_{O4}^- \frac{R_7}{R_7 + R_8}.$$

inverting comparator with hysteresis centered around zero and the threshold levels:

The voltage swing at the output of the circuit (V^+_{O4}, V_{+O4}), is a function of the limiter comprising resistor R_9 and diodes

$$V_{I1} > V_{I2} \exp\left[\frac{V_H^+}{V_T} \bullet \frac{R_3}{R_4}\right], \text{ or } V_{I1} < V_{I2} \exp\left[\frac{V_H^-}{V_T} \bullet \frac{R_3}{R_4}\right].$$

FIGURE 1



Signal-variant hysteresis allows you to compare signals with a constant-ratio trip window for a wide range of signal levels.

D_3 to D_8. The output voltage, V_{O4}, changes state when V_{O3}>V^+_H (output goes low) or V_{O3}<$V^=_H$ (output goes high). These conditions correspond to

The design allows you to distinguish between two signals, with a tolerance set by the external components, R_3, R_4, R_7, and R_8. When V^+_{O4}=|V^-_{O4}|, the tolerance is symmetrical. With the given values, the output state changes when one input signal is approximately 30% higher than the other. If you need to use the circuit over a wide temperature range, you should take the thermal dependence of the threshold levels into consideration. The circuit uses an LM324 quad op amp. If you use a rail-to-rail amplifier (for example, the LMC6484), the output limiter is unnecessary. (DI #2166) EDN

To Vote For This Design, Circle No. 416

Simple LCD interface takes two wires

ED MASTE, JEM DESIGNS, PICKERING, ON, CANADA

Alphanumeric LCD modules can provide an attractive display for a project, but their parallel I/O lines require a large number of outputs from a µC. For example, a direct LCD interface would consume all of the I/O pins of a small µC, such as the eight-pin PIC12C508 from Microchip Technology (Chandler, AZ).

However, the circuit in **Figure 1a** implements a clocked serial input for an LCD module and allows a µC to communicate with the LCD over just two signal lines. You can fit the circuit onto a small pc board and mount the board directly behind the LCD module. The connection between the board and LCD module comprises just four wires, including V_{CC} and ground.

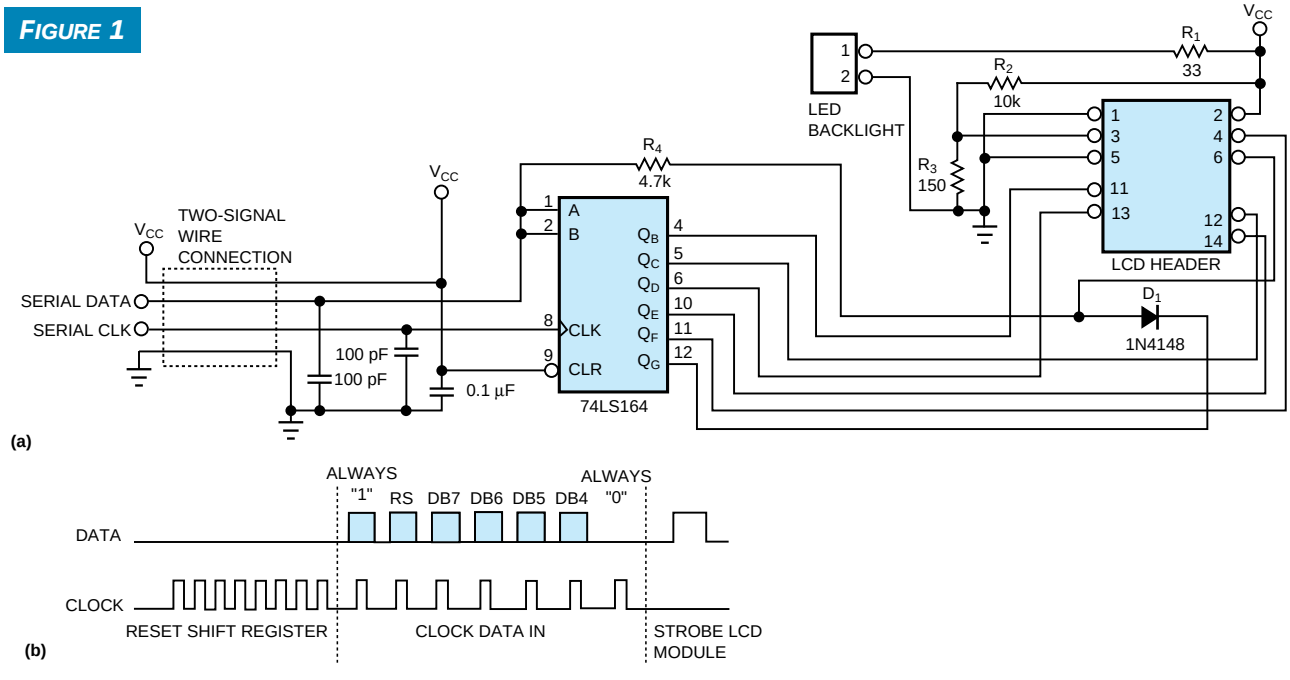
A 74LS164 serial-to-parallel shift register forms the heart of the circuit and communicates with the LCD module in 4-bit mode. The shift register's outputs directly drive the LCD module's data inputs, DB7 through DB4, as well as RS, the control/data select input. The only signal that is not a direct output from the shift register is the LCD module's enable pin, which is Pin 6. R_4 and D_1 derive the enable input and ensure that the enable signal remains low until valid data is present at all other outputs. R_1 is a current-limiting resistor for an LED backlight if the LCD module has one, and R_2 and R_3 set the contrast level of the module. You can replace R_2 and R_3 with a potentiometer if an adjustable contrast level is desirable.

The µC must first clock in at least seven zeros for the first write (**Figure 1b**). This series of zeros guarantees that output Q_G of IC_1 is low and will remain so for the next six clock cycles. This low level ensures that the enable input (Pin 6 of the LCD module) remains low because of diode D_1. The serial-input stream then must consist of a one, followed by RS, and finally DB_7 through DB_4. With the data input low, an additional clock pulse shifts in a zero and lines up all of the outputs to the LCD module with the correct shift-register pins. The first clocked-in one now appears at Q_G, which reverse-biases D_1. The enable continues to remain low because the data input to the circuit is low, and this situation pulls the enable low through R_4. The data input then must go high for at least 450 nsec without a clock pulse to provide the enable pulse for the LCD module. This process, including clocking in at least six zeros, is then repeated for the next 4-bit write.

For further information, including minimum cycle times and 4-bit-mode communication, consult the technical documentation provided by an LCD module manufacturer, such as Optrex (Plymouth, MI). (DI #2177) EDN

To Vote For This Design, Circle No. 417

FIGURE 1



Using a 74LS164 shift register and 4-bit communication, a µC can control an LCD module using just two lines: serial data and clock (a). The µC must first clock a series of zeros to the circuit, followed by a one, RS, and DB_7 through DB_4 (b).

Inexpensive logic controls stepper motor

DAVID ELLIS, ELLIS LINDAUER, PULLMAN, WA

A number of sophisticated ICs for stepper-motor control are now available. However, the advanced features of these chips—self-clocking, high-current drive, and full-step, half-step, and direction control—are often unnecessary or remain unused. For a design that needs to control only the number of steps, drive speed, and direction, you can make a very simple and inexpensive driver using two low-level logic chips (**Figure 1**). The cost of this controller is less than $1; the cost of dedicated motor-control ICs starts at around $5. The drawback is a slight increase in board space.

TABLE 1—STEPPER-MOTOR-DRIVE SEQUENCE

CCW ↑

State	A	B	C	D
1	ON	OFF	ON	OF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF

(a)

↓ CW

State	A	B	C	D
1	1	0	1	0
2	1	0	0	1
3	0	1	0	1
4	0	1	1	0

(b)

Going back to the basics, you can control a standard stepper-motor drive, whether bipolar or unipolar, using a four-step sequence (**Table 1a**). By replacing the on and off states with ones and zeros, respectively (**Table 1b**), Column B becomes the logical inverse of Column A, and Column D becomes the logical inverse of Column C. Thus, the corresponding state diagram (**Figure 2**) comprises just 2 bits. Clockwise rotation results from using a logical one to move sequentially from state one to state four and back to state one. Likewise, counterclockwise rotation results from using a logical zero to move through the states in the reverse order.

You can then produce the present-state/next-state assignment (**Table 2**) and the next-state maps (**Figure 3**). Then, by inspection, the logical choice is to loop the state maps out for D flip-flops, which produces the following two logic equations:

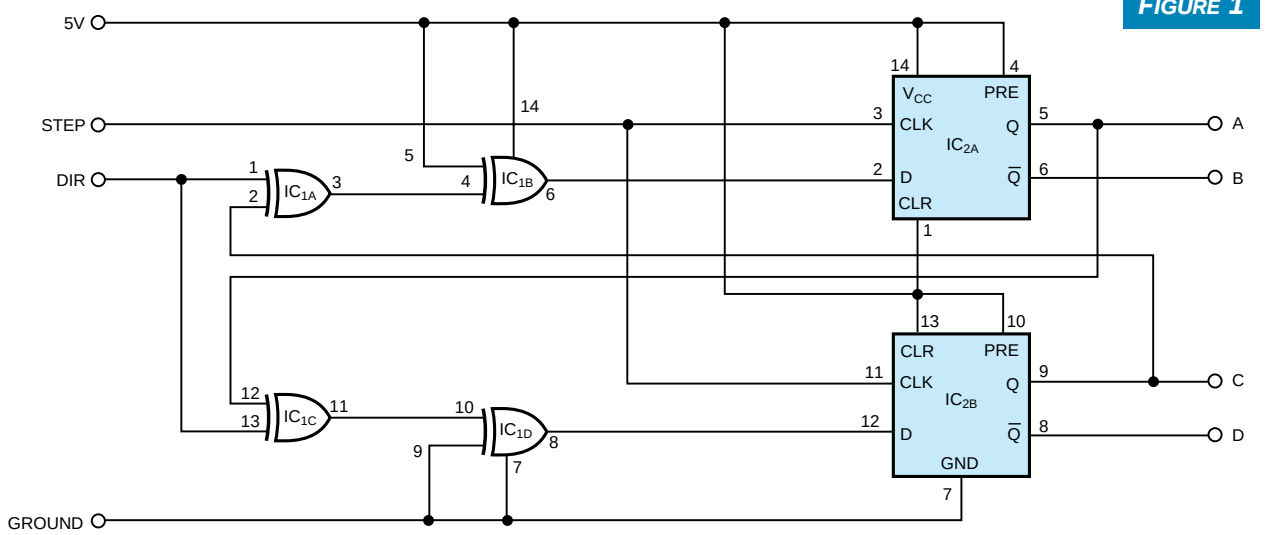
$$DA=((DIR)(C))+((DIR)(C)); \quad (1)$$

$$DB=((DIR)(A))+((DIR)(A)). \quad (2)$$

Equation 1 is an exclusive NOR, and **Equation 2** is an exclusive OR. To save space, you can use a single quad XOR chip to implement both equations. A dual D flip-flop completes the logic driver, as **Figure 1** shows. Using rising-edge-triggered D flip-flops helps keep the design simple while eliminating mode-change faults.

The circuit derives the four outputs from the Q and Q

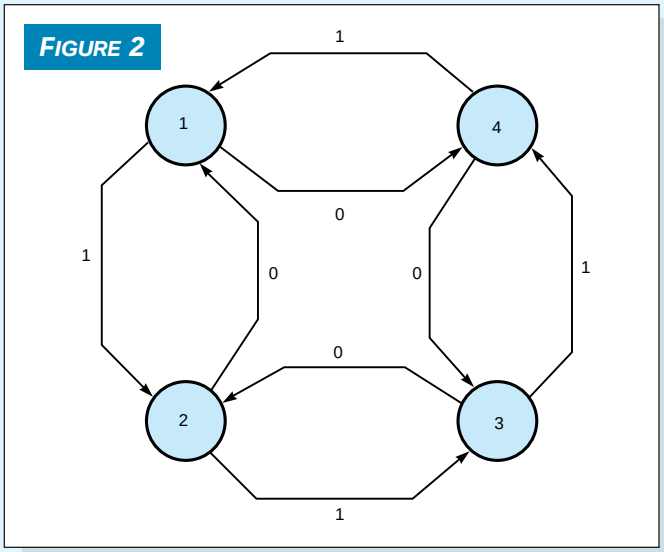
FIGURE 1



NOTES:
IC_1=74LS36;
IC_2=74LS74.

Two logic-level ICs can implement simple and inexpensive control of a stepper-motor driver.

FIGURE 2



The state diagram for the stepper-motor controller comprises just 2 bits.

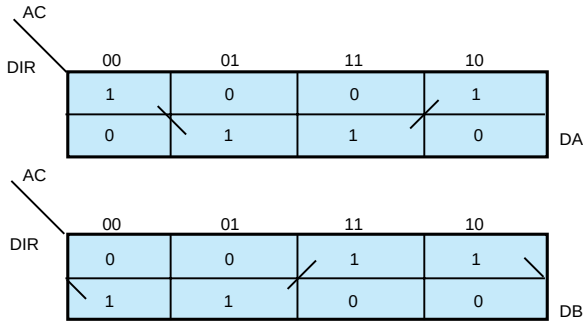
outputs of D flip-flops IC_{2A} and IC_{2B} in **Figure 1**. IC_{1A} XORs the Q output of flip-flop IC_{2B} with the DIR input, and the circuit transforms the output into an XNOR by using IC_{1B} as a controlled inverter. IC_{1B} then drives the D input of flip-flop IC_{2A}. Similarly, IC_{1C} XORs the Q output of IC_{2A} with the DIR input. The output of IC_{1C} drives XOR IC_{1D}, which acts as a noninverting buffer. The output of IC_{1D} drives the D input of IC_{2B}. Using XOR gate IC_{1D} as a buffer keeps the propagation delays to the D inputs of the flip-flops equal, which helps the circuit avoid any race conditions. The STEP signal is the step-rate input, which drives the clock inputs of both flip-flops.

The last design task is to add the appropriate-sized transistors to drive the stepper motor. In the case of the unipolar motor, output signals A, B, C, and D can directly drive the transistors. To drive a bipolar motor, you can use the A and C outputs to drive one-half of two H-bridges and the B and D outputs to drive the other corresponding half of the H-bridges. This design is possible because the B output is the inverse of A, and D is the inverse of C. (DI #2176)

EDN

To Vote For This Design, Circle No. 418

FIGURE 3



The next-state maps correspond to two simple logic equations.

TABLE 2—PRESENT-STATE/NEXT-STATE ASSIGNMENT TABLE

Present state			Next state	
A	C	DIR	A	C
0	0	0	1	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

Low-voltage reset operates below 2.7V

BOB KELLY, MAXIM INTEGRATED PRODUCTS, SUNNYVALE, CA

New personal digital assistants, pagers, and other battery-powered systems operate at or below 2.7V, but power-on resets with thresholds below 2.6V are not commonly available. You can resolve this problem using a circuit that combines a 1.2V reference and a micropower regulator (**Figure 1a**). IC_1 integrates these two functions in a tiny SOT-143 package. A power-on-reset function must become active before the supply voltage reaches its nominal value, and IC_1 operates properly for supply voltages above 1.21V.

The R_1/R_2 divider and internal 1.204V reference establish a threshold that determines when the circuit asserts an active-low at the output. For the values in the **figure**, this threshold is 2.25V (**Figure 1b**). IC_1 has an open-drain output, so R_3 and C_1 control the length of the active-low pulse, RESET. In this case, the pulse length, or reset interval, is approximately 54 msec, which is sufficient reset time for

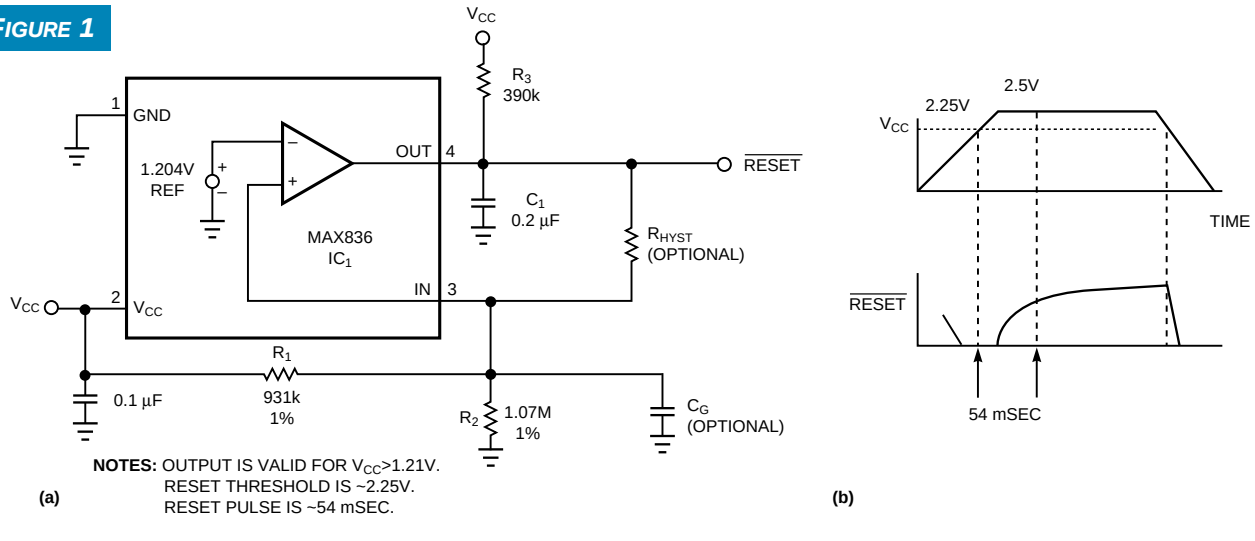
most µCs and other digital circuits.

Low power consumption distinguishes this circuit. The IC typically draws only 5 µA, and the R_1/R_2 divider draws slightly more than 1 µA in a 2.7V application. Pullup resistor R_3 consumes power only when the supply voltage droops out of tolerance, so the power loss is minimal in normal operation.

To prevent erratic behavior, IC_1 offers approximately 6 mV of built-in hysteresis. For more hysteresis, you can add a large-value resistor, R_{HYST}, between the IC's input and output; to reject short transients, IC_1 has an inherent glitch immunity of 35 µsec with 100 mV of overdrive. The input capacitance works with R_1 and R_2 to provide some lowpass-filter action. For further immunity from transients, which is unnecessary unless the power bus is noisy, you can form an additional lowpass filter by adding a small-value capacitor, C_G, to the input pin. (DI #2174) EDN

To Vote For This Design, Circle No. 419

FIGURE 1



A 1.2V reference and micropower regulator in IC_1 (a) provide an active-low reset pulse of approximately 54 msec at power-up or when V_{CC} dips below 2.25V (b).

Alternating LED blinker uses four parts

ANDY MENG, CINCINNATI, OH

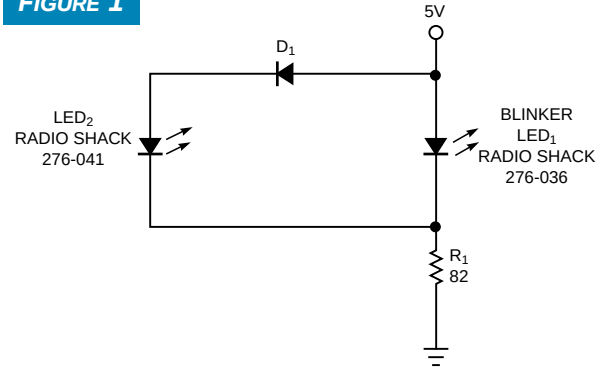
The circuit in **Figure 1** is an easy-to-make attention-getter and runs for a week or longer on two AA cells. In September, I used this circuit for a school fundraiser, and it helped me generate more than $100. My dad showed me a circuit in *EDN* that did the same thing, but it uses more parts (see "Alternating LED flasher uses minimal parts," *EDN*, Nov 20, 1997, pg 104).

The main element of this circuit is LED_1, a Radio Shack 276-036 blinking red LED. D_1 can be almost any silicon diode. The forward bias of D_1 brings the turn-on voltage of LED_2 up to 2.5V. R_1 is a current-limit resistor for LED_1, and this resistor also reduces the current of LED_1 for longer battery life.

LED_2 is a Radio Shack 276-041 red LED. When you apply power, LED_1 turns on and drops the voltage across LED_2 to 1V. When LED_1 turns off, the voltage across LED_2 equals 3V, and LED_2 turns on. (DI #2172) EDN

To Vote For This Design, Circle No. 420

FIGURE 1



This blinking-LED circuit, designed by a seventh grader, uses only four parts.

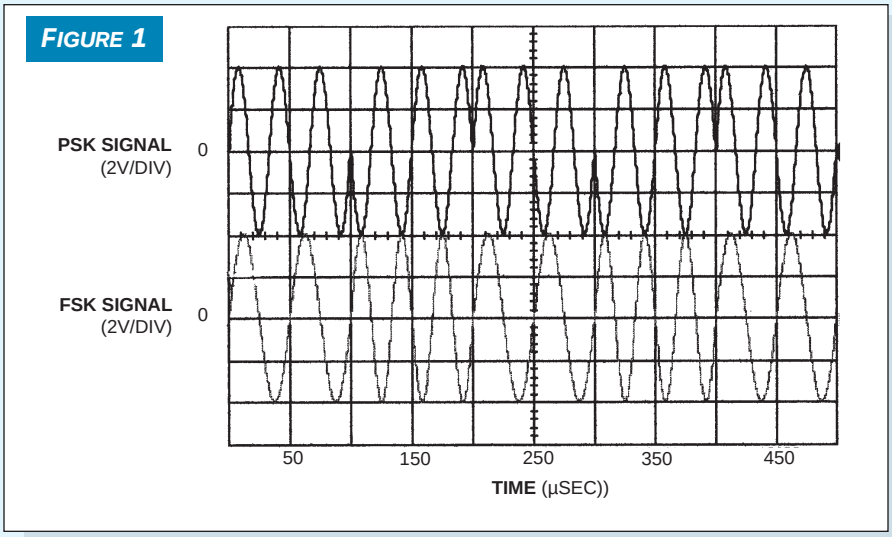
Spice generates PSK and FSK signals

DEBRA HORVITZ, GALAHAD SYSTEMS, LAGUNA HILLS, CA

Creating generators for amplitude-, frequency-, and phase-modulated signals can greatly simplify communication-system simulation. Although Spice includes a basic set of waveform generators, it includes no built-in support for many types of signals. You must create these signals from combinations of elements, and you can create variations of these built-in generators using Spice 2-dependent sources. However, using dependent sources to generate complex waveforms can require fairly complex Spice subcircuits.

Fortunately, the nonlinear, arbitrary dependent source—the B element—in Berkeley Spice 3 and IsSpice4 (Intusoft, San Pedro, CA) provides a quantum leap in capability over Spice 2-dependent sources. The B element is more versatile and easier to use. For example, **Listing 1** is the IsSpice4 subcircuit for a parameterized phase-shift-keying (PSK) source; **Listing 2** is the subcircuit of a parameterized frequency-shift-keying (FSK) source. **Figure 1** shows the resultant output signals of each subcircuit generator.

FIGURE 1



The behavioral and mathematical capabilities of Spice 3 and IsSpice 4 make it easy to create PSK and FSK signals.

The PSK subcircuit produces a coherent binary PSK signal according to the following equations:

$$s_1(t) = \sqrt{\frac{2 \bullet EB}{TB}} \bullet \sin(2\pi ft),$$

$$s_2(t) = -\sqrt{\frac{2 \bullet EB}{TB}} \bullet \sin(2\pi ft),$$

where EB is the transmitted energy per bit, TB is the bit duration, and f is the transmission frequency equal to NC/TB. (NC is an integer constant, and the period is 2×TB; thus, the duty cycle equals 50%.) In **Listing 1**, the input voltage source, VSIG, produces the polar form of the input signal. Using the B1 element, the subcircuit multiplies this input by the local oscillator voltage, VLO1, to produce the PSK output signal.

The subcircuit, FSK, produces a coherent binary FSK signal according to the following equations:

$$s_1(t) = \sqrt{\frac{2 \bullet EB}{TB}} \bullet \sin(2\pi f_1 t),$$

$$s_2(t) = \sqrt{\frac{2 \bullet EB}{TB}} \bullet \sin(2\pi f_2 t).$$

where f_1 is the high-bit transmission frequency, and f_2 is the low-bit transmission frequency. The frequency, f_1, is equal to NC+1/TB. The frequency, f_2, is equal to NC+2/TB. For simplicity, two pulse generators, VSIG and VSIGN, produce the input signal, m(t), and the inverse of the message signal, M(t), respectively. Again, using the B1 element, the subcircuit code multiplies these signals by the appropriate frequency generator: VLO1 for logic high and VLO2 for logic low. The sum of the resultant signals produces the FSK output signal. (DI #2173) EDN

LISTING 1—ISSPICE4 SUBCIRCUIT FOR PSK SOURCE

```
.SUBCKT PSK  3   ; Signal Generator for PSK
VSIG 2 0  PULSE  {-(EB)^2}  {(EB)^2}  {TD}  {TR}  {TF}
+ {50/100*(2*TB-TR-TF)}  {2*TB}
B1 3 0 V = V(1) * V(2)
VLO1 1 0  SIN 0 {(2/EB)^2}  {(NC)/TB}
.ENDS
```

LISTING 2—SUBCIRCUIT FOR FSK SOURCE

```
.SUBCKT FSK  5 ; Signal Generator for FSK
VSIG 3 0  PULSE  0  {(EB)^2}  {TD}  {TR}  {TF}
 {50/100*(2*TB-TR-TF)}  {2*TB}
VSIGN 4 0  PULSE  {(EB)^2}  0  {TD}  {TR}  {TF}
 {50/100*(2*TB-TR-TF)}  {2*TB}
B1 5 0 V = (V(1) * V(3)) + (V(2) * V(4))
VLO1 1 0  SIN  0  {(2/EB)^2}  {(NC+1)/TB}
VLO2 2 0  SIN  0  {(2/EB)^2}  {(NC+2)/TB}
.ENDS
```

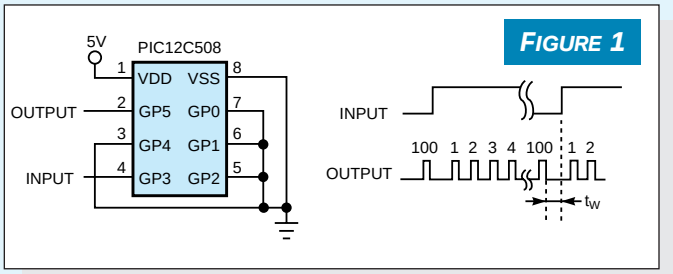
Frequency multiplier improves line readings

YONGPING XIA, TELDATA INC, LOS ANGELES, CA

Because of the low frequencies involved, accurately measuring line-frequency variations is complicated. When you use an ordinary frequency counter with a 1-sec gate time, the reading would be 59, 60, or 61 Hz. To obtain 0.01-Hz accuracy, you must increase the gate time to 100 sec, a scale that most frequency counters do not offer. Moreover, reading updates are slow with this scale choice. One way to improve reading accuracy without sacrificing update time is to use a frequency multiplier. For instance, a frequency counter with a 1-sec gate time could provide 0.01-Hz accuracy if you multiply the line frequency by 100. Traditionally, you would configure the multiplier by using a VCO, a PLL, a frequency divider, and a lowpass filter. **Figure 1** shows another method, using only one component, to multiply the line frequency.

The PIC12C508 is an eight-pin, low-cost µC. It has a built-in 4-MHz oscillator and a reset circuit; it thus needs no external components. One of the µC's unique features is its software-based clock-frequency calibration. The controller dedicates one register, OSCCAL, to the calibration function. The upper 4 bits of OSCCAL accommodate 16 steps of calibration. Each step changes the clock frequency approximately 1.6%. The maximum change is approximately 25%. The line signal routes to the µC's GP3 pin. When the controller detects a low-to-high change in GP3, the program generates 100 pulses on GP5 and ignores the condition of GP3 (**Listing 1**). It then rechecks the status of GP3. A high value for GP3 means that the pulse generation is too slow; OSCCAL then increases by one step. If GP3 is low, then a counter operates until GP3 goes high.

FIGURE 1



An inexpensive µC, using no external components, allows you to measure line frequency with 0.01-Hz accuracy.

The number in the counter represents the delay (t_W) between the last pulse and the next line-signal change. The µC compares t_W with a fixed number. If t_W is larger than that number, indicating that the clock frequency is too high, the value in OSCCAL decreases by one step to slow down. If t_W is smaller than the fixed number, the value in OSCCAL increases by one step to speed up. Thus, the automatic clock-frequency adjustment continuously sends a signal at 100 times the line frequency. Note that, because of the calibration range, this approach has a limited input-frequency range. However, it accommodates line-frequency measurements from 58 to 62 Hz. You can download the assembly code from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2182. (DI #2182) EDN

To Vote For This Design, Circle No. 422

LISTING 1—PIC12C508 CODE FOR LINE-FREQUENCY MEASUREMENT

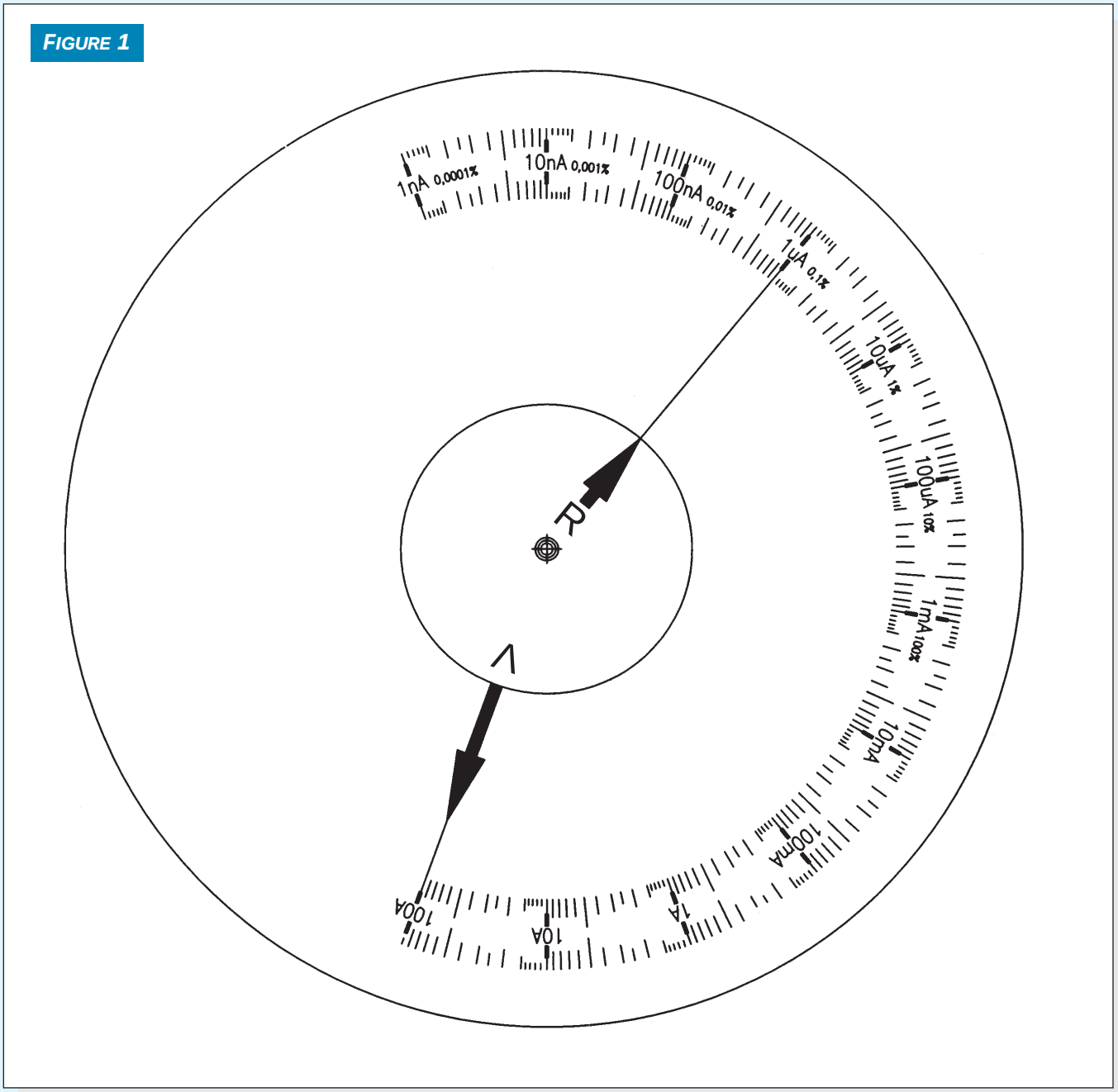
```
        LIST    P=12C508

status          equ     0x03            ;
osccal          equ     0x05            ;
gpio            equ     0x06            ;
                                        ;
C               equ     0               ;
Z               equ     2               ;
                                        ;
cnt_1           equ     0x07            ;
cnt_2           equ     0x08            ;
cnt_3           equ     0x09            ;
cnt_4           equ     0x0a            ;
temp            equ     0x0b            ;
                                        ;
                org     0x0             ;
                                        ;
                movlw   0x80            ;
                movwf   osccal          ;
                movlw   0x1f            ;
                tris    gpio            ;
lp_1            btfsc   gpio, 3         ;       wait input low
                goto    lp_1            ;
lp_2            btfss   gpio, 3         ;       wait input high
                goto    lp_2            ;
lp_3            movlw   0x64            ;
                movwf   cnt_1           ;
lp_4                                    ;
                bsf     gpio, 5         ;       output high
                call    dly             ;
                nop                     ;
                nop                     ;
                bcf     gpio, 5         ;       output low
                decfsz  cnt_1, 1        ;
                goto    next_1          ;
                goto    last_cycle      ;
next_1          call    dly             ;
                goto    lp_4            ;
                                        ;
last_cycle                              ;
                clrf    cnt_3           ;
                clrf    cnt_4           ;
lp_5            incfsz  cnt_3, 1        ;
                goto    next_2          ;
                incf    cnt_4, 1        ;
next_2          btfss   gpio, 3         ;       wait input high
                goto    lp_5            ;
                movf    cnt_4, 0        ;
                btfss   status, Z       ;
                goto    freq_down       ;
                movlw   0x30            ;
                subwf   cnt_3, 0        ;
                btfss   status, C       ;
                goto    freq_up         ;
                goto    freq_down       ;
                                        ;
freq_down                               ;       decrease frequency by 1.6%
                movlw   0xf0            ;
                andwf   osccal, 1       ;
                btfsc   status, Z       ;
                goto    lp_3            ;
                movlw   0x10            ;
                subwf   osccal, 1       ;
                goto    lp_3            ;
                                        ;
freq_up                                 ;       increase frequency by 1.6%
                movlw   0xf0            ;
                andwf   osccal, 1       ;
                movlw   0x10            ;
                addwf   osccal, 1       ;
                btfss   status, Z       ;
                goto    lp_3            ;
                subwf   osccal, 1       ;
                goto    lp_3            ;
                                        ;
dly                                     ;       delay loop
                movlw   0x18            ;
                movwf   cnt_2           ;
dly_lp          decfsz  cnt_2, 1        ;
                goto    dly_lp          ;
                retlw   0x00            ;
                                        ;
                end                     ;
```

Circular slide rule provides quick results

JACEK PAWLOWSKI, PW INMEL, ZIELONA GORA, POLAND

In analog-circuit design, most calculations you make need not be very precise. If you need an LED-current calculation or a coupling-capacitor value, for example, ±5% or even ±10% accuracy is usually adequate. It's sometimes inconvenient to make these calculations with a pocket calculator. For example, finding the cutoff frequency of a 3.3-kΩ/47-pF network requires approximately 20 key presses. The circular slide rule using the patterns in **Figures 1** through **4** simpli-

FIGURE 1

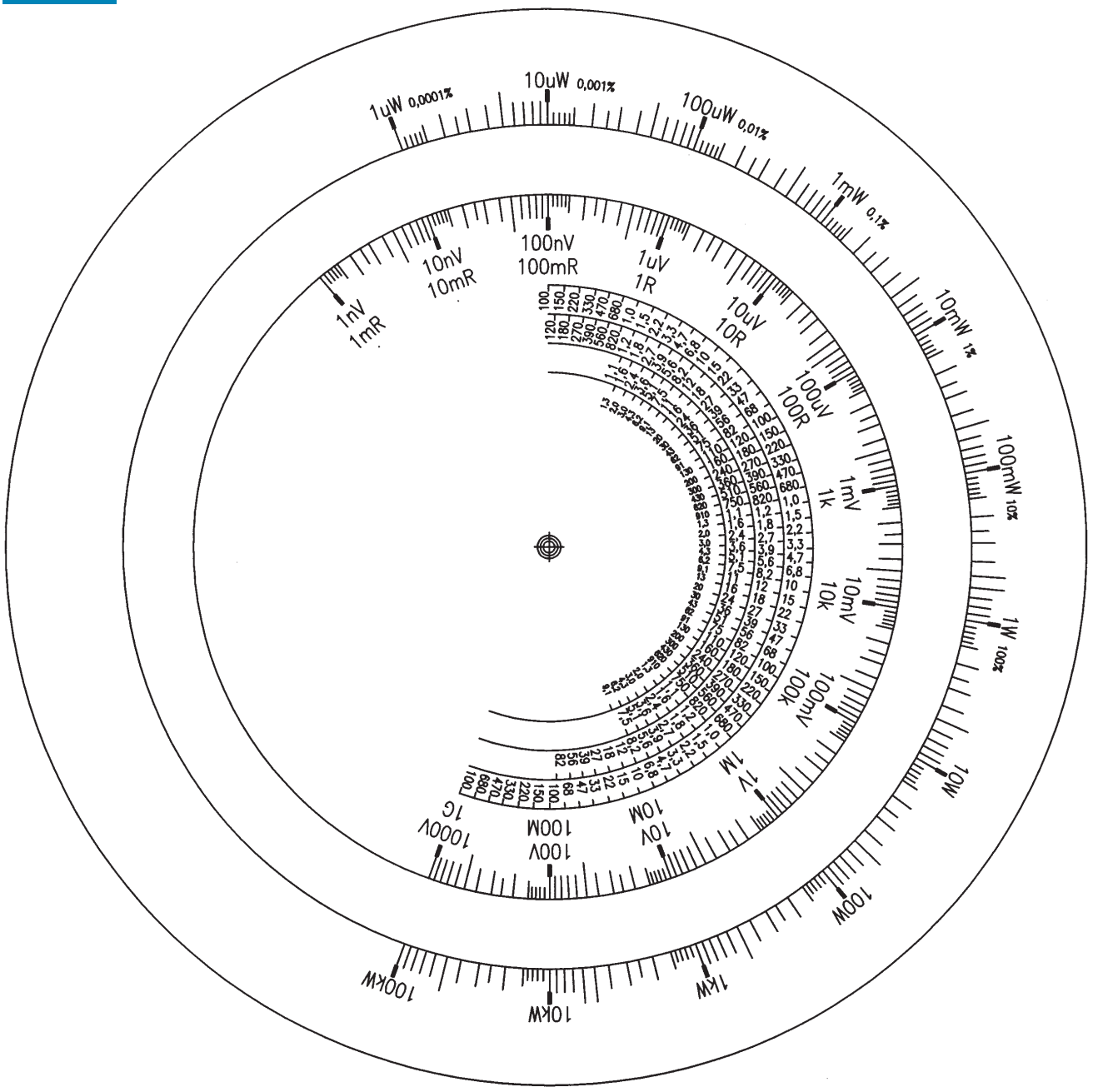


This wheel, using transparent material, gives current values of 1 nA to 100A.

fies such calculations. The slide rule uses one large, opaque, double-sided wheel and two smaller, transparent wheels.

You can make the opaque wheel by gluing back-to-back the patterns in **Figures 2** and **4**. For the transparent wheels, you simply photocopy the patterns in **Figures 1** and **3** onto overhead-transparency foils. To attach the wheels, you can use a rivet or a screw and nut. Using side A (**Figures 1** and **2**), you can calculate resistance (V/I), power (VI), and percentage products ($\Delta=\delta \cdot X$, where δ is a percentage). X can be voltage, current, power, or resistance. Side A also gives stan-

FIGURE 2



This wheel, an opaque background, gives voltage, current, power, and dissipation-factor percentages. The inner rows give standard EIA resistance values.

dard EIA values for resistance.

$$\tau = RC,$$

$$F_C = \frac{1}{2\pi RC},$$

$$X_C = \frac{1}{2\pi fC},$$

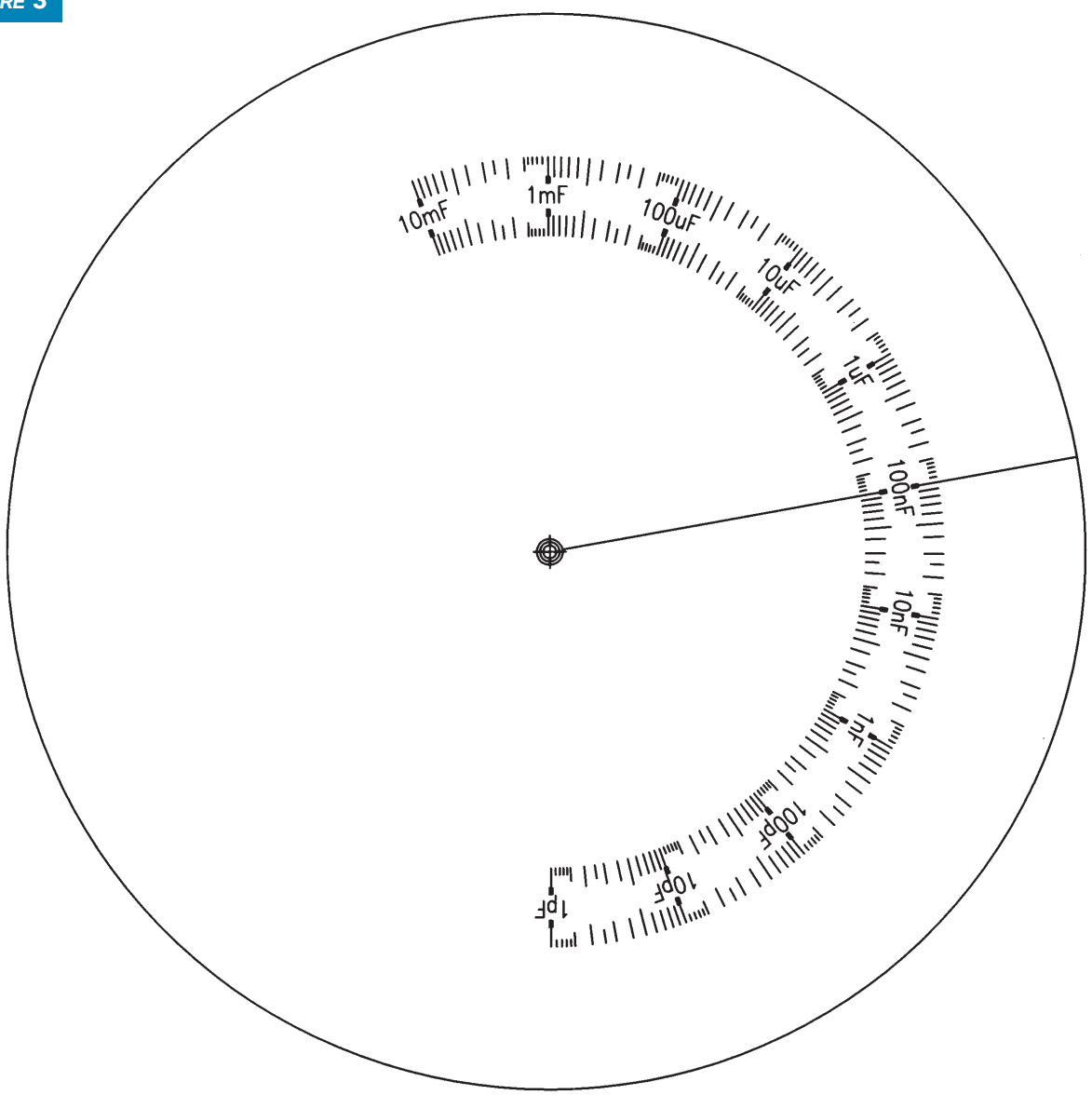
$$F_{RES} = \frac{1}{2\pi\sqrt{LC}},$$

$$T = \frac{1}{f}.$$

On side B (**Figures 3** and **4)**, you can calculate the following formulas:

$$F_{RES} = \frac{1}{2\pi\sqrt{LC}},$$

FIGURE 3



This wheel, copied onto transparency material, gives capacitance values from 1 pF to 10 mF (10,000 µF).

$$L = \frac{1}{4\pi^2 (F_{RES})^2 C},$$

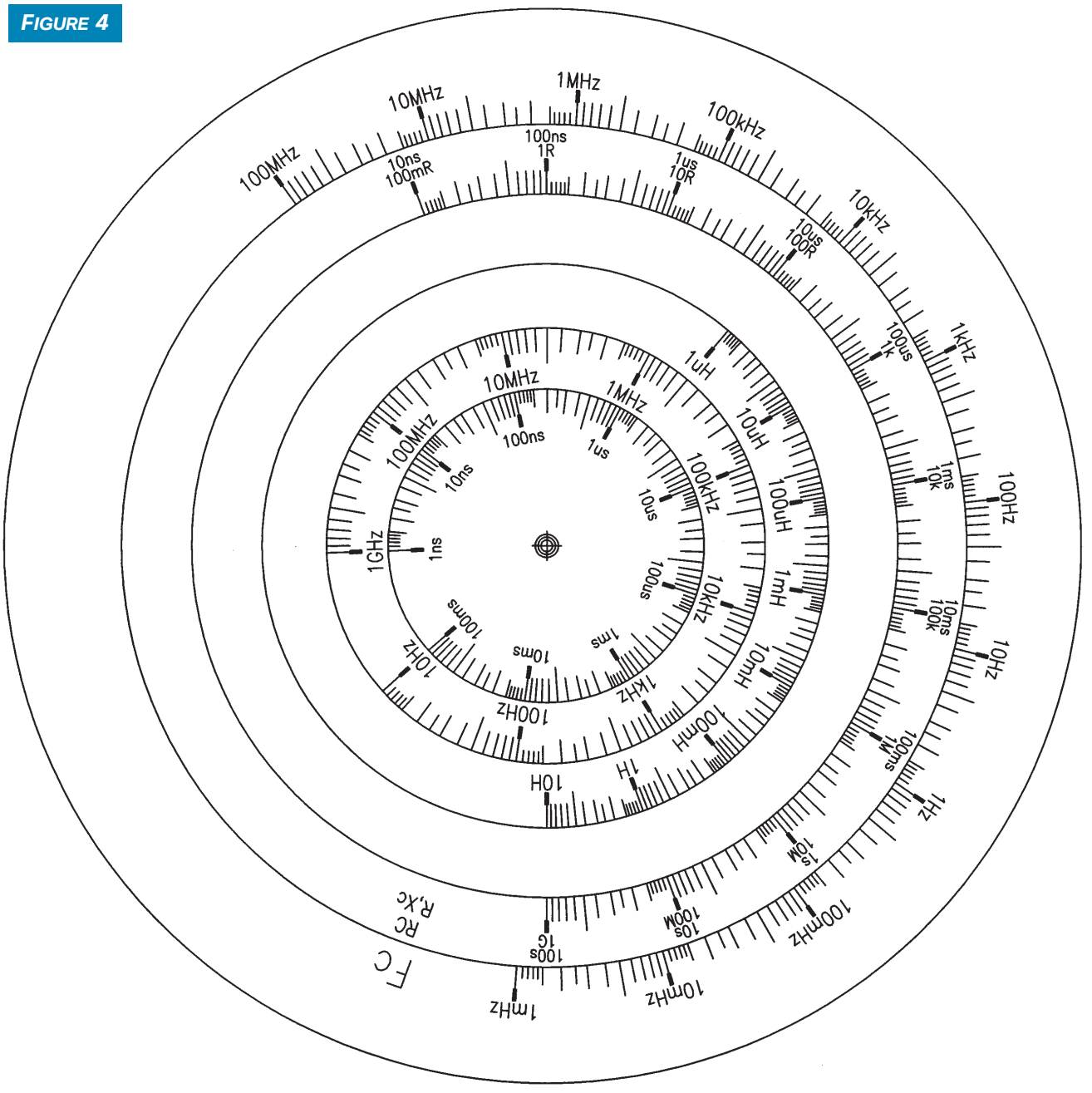
Any quantity can be the unknown. For example, you can calculate

$$C = \frac{1}{4\pi^2 (F_{RES})^2 L}.$$

You can modify the slide rule to incorporate the formulas you use most often. The design of the slide rule uses AutoCAD LT. (DI #2137) EDN

To Vote For This Design, Circle No. 423

FIGURE 4



This wheel, the opaque backing for the wheel in Figure 2, relates resistance, frequencies, and time constants to the capacitance values on the wheel in Figure 3.

Supervisory circuit monitors modem connection

J BASILIO SIMÕES AND JORGE LANDECK, INSTRUMENTATION CENTER OF THE UNIVERSITY OF COIMBRA, COIMBRA, PORTUGAL

In telemetry or security applications in which a modem connection automatically establishes itself between two systems, a failure in one of the systems can interrupt the data exchange while the connection remains established until someone physically breaks the line or restarts the failed system. This situation can also happen if your PC crashes during a long-lasting Internet download. To prevent this problem, the circuit in **Figure 1** continuously supervises the RS-232C data lines (TxD and RxD) and automatically hangs up the connection when the circuit detects a long period without transmitted or received data.

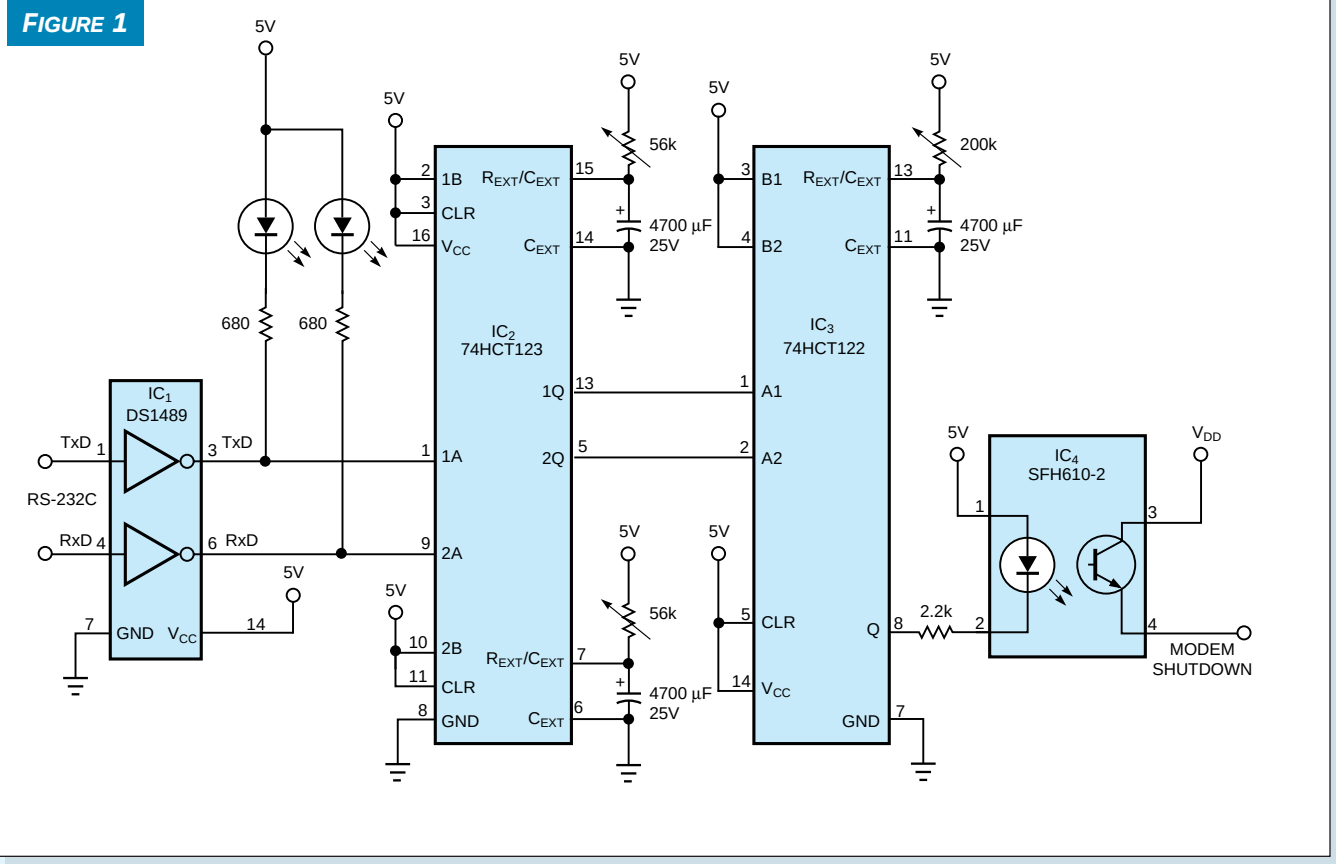
The TxD and RxD inputs to the circuit (pins 2 and 3 on the D-type RS-232C connector) first drive line receiver IC_1, whose outputs then drive the two inputs of IC_2's dual retriggerable monostable multivibrator (one shot). When the system establishes a new connection, the first exchange of data triggers IC_2, and positive pulses of duration $T_X=0.45\times R_X\times C_X$ appear at the 1Q and 2Q outputs. Each new transition on the data lines retriggers the associated one shot, which extends the output pulse for a new period, T_X. In this application, R_X=56 kΩ, and C_X=4700 µF corresponds to a period of about 2 minutes.

If a problem arises with one of the systems and it stops transmitting data for a period longer than T_X, the corresponding one shot's output pulse goes low, indicating that the system should hang up the connection. To allow for the self-restarting of the system, another one shot, IC_3, shuts down the modem for a period T_Y, determined by the associated timing resistor and capacitor.

The conditioning circuit for the output signal uses an optocoupler, IC_4, to drive the shutdown input of the modem. If your modem has no shutdown input, you can solve the problem by powering the modem through a relay driven by IC_4. (DI #2186) EDN

To Vote For This Design, Circle No. 526

FIGURE 1



A lack of activity on the TxD or RxD lines fails to retrigger IC_2, which causes the circuit to hang up the modem connection.

Circuit optimizes phototransistor bandwidth

DAVID MAGLIOCCO, CDPI, SCIENTRIER, FRANCE

A simple circuit can improve the dynamic performance of a phototransistor for use in low- to medium-speed applications as fast as 100 kbps, such as optical isolation of an RS-232C serial line (**Figure 1**). In low-cost applications that require high voltage insulation, low part count, and low power, you can use consumer components, such as Siemens' SFH421 IR LED with an SFH320 IR phototransistor. The rise/fall-time rating of the LED is 500 nsec, which is fast enough, but the phototransistor's rise/fall time is 5 µsec for the fastest version and 8 µsec for the slowest with a 1-kΩ load.

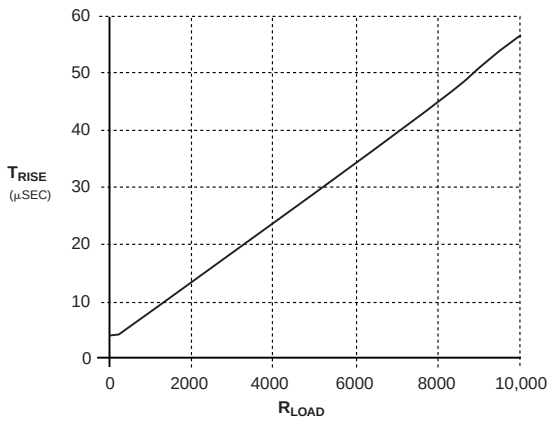
Phototransistors have less favorable dynamic behavior than do photodiodes because, in addition to the collecting and charging processes, phototransistors also experience a delay stemming from the amplification mechanism (Miller effect). For the rise and fall time of a phototransistor, the following relationship applies:

$$t_{r,f} = \sqrt{(1/2f_T)^2 + b(R_{LOAD} \bullet C_{CB} \bullet V)^2},$$

where f_T is the transition frequency, C_{CB} is the collector-base capacitance, V is the gain, and b is a constant whose value lies between 4 and 5. For R_{LOAD}=1 kΩ, 4<b<5, 100<V<1000, C_{CB}≈2.5 pF, and f_T≈100 kHz, the result is 5 µsec<$t_{r,f}$<8 µsec.

A plot of rise time vs load resistance provides a more practical way to evaluate the maximum transmission speed of **Figure 1a**'s circuit without using additional compensation circuitry (**Figure 2**). For a 2400 bps serial link with less than 10% error on the bit length, the rise-time estimate is 42 µsec, which corresponds to a 7.5-kΩ resistor.

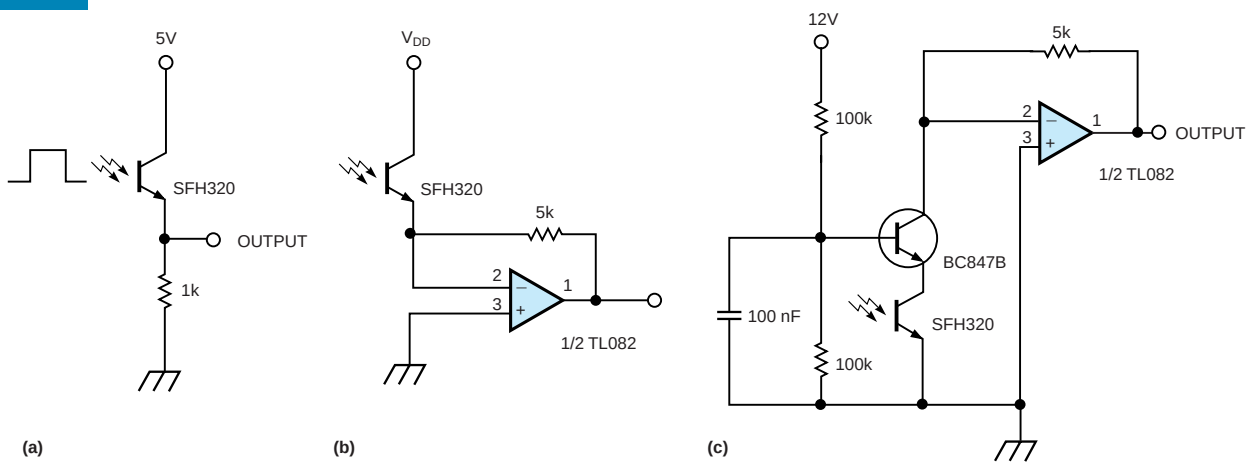
FIGURE 2



The relationship between worst-case rise time and load resistance for the basic phototransistor circuit indicates that for a t_{RISE} of 42 µsec, R_{LOAD} should equal 7.5 k .

To get a faster response time, you can use an op amp, but

FIGURE 1



A basic phototransistor (a) has a 5- to 8-µsec rise/fall time with a 1-k resistor. An op amp can improve a phototransistor's response time (b) but performs poorly without additional phase compensation. Alternatively, you can use a transistor to isolate the phototransistor's capacitance and to hold the signal voltage across the phototransistor to a stable dc value (c).

without phase compensation the result is poor. In **Figure 1b**'s circuit, the phototransistor's internal capacitance and the amplifier's gain bandwidth limit the overall speed. To cancel the side effect of the phototransistor's C_{CB}, the amplifier needs a small capacitor in parallel with R. Choosing the right value for the compensation capacitor is a difficult task, because the current-to-voltage converter exhibits a two-pole response. Also, to ensure stability, you need to consider phase compensation and bandwidth together.

Fortunately, there is another way to get the best of the phototransistor bandwidth. You can isolate the phototransistor's internal capacitance with a transistor (**Figure 1c**). The transistor, with its low output impedance and its large gain bandwidth, holds the signal voltage across the phototransistor to a stable dc value. The versatile BC847 can do the job, and, with Siemens' SFH320, the circuit can reach a transmission speed of 153.6 kbps. You can't increase the speed beyond this point, even with the fastest op amp, because of the photoelectric delay that the charging and collecting processes cause. (DI #2188) EDN

To Vote For This Design, Circle No. 527

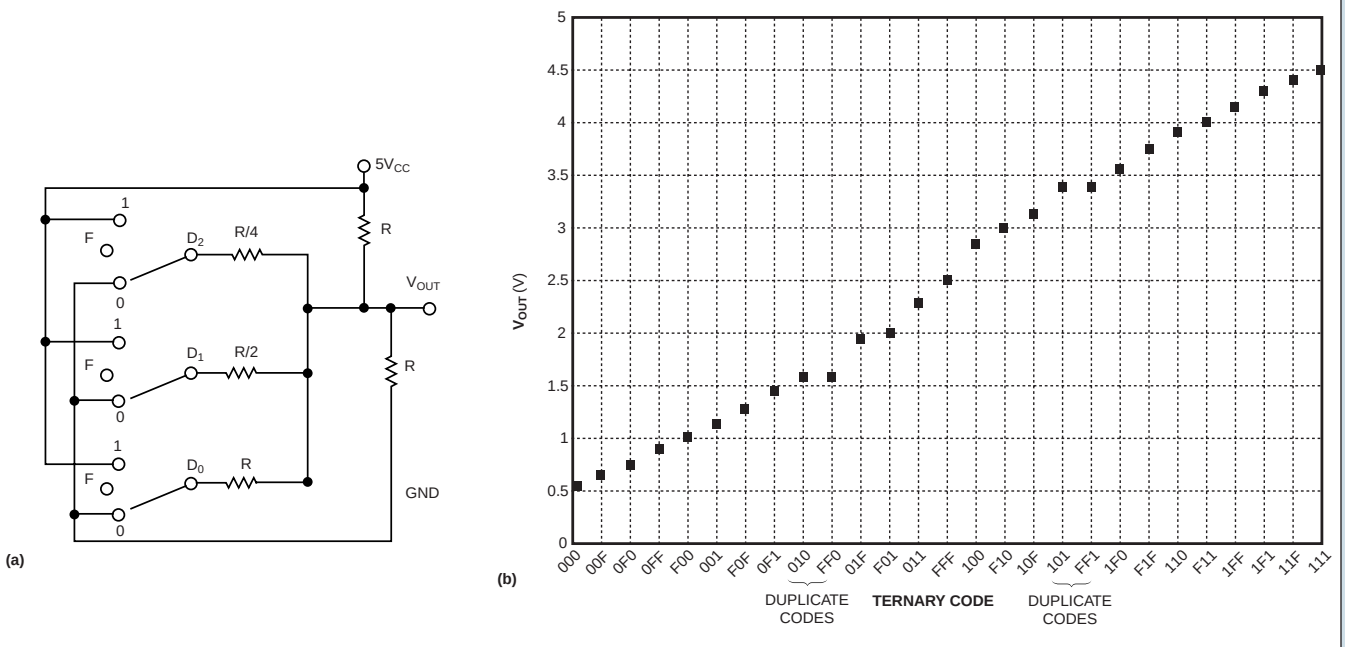
Technique increases low-cost DAC's resolution

JEREMY DEAN, THOMSON-THORN MISSILE ELECTRONICS LTD, BASINGSTOKE, UK

Cost-sensitive µC applications often employ resistor chains to implement crude DACs. You can extend this method by exploiting the way in which many µCs allow individual output pins to be set to either low ("0"), high ("1"), or floating ("F") states. A converter can thus respond to ternary rather than binary codes.

The resistive network in **Figure 1a** has three inputs. Allowing each input to be either 0, 1, or F results in the transfer characteristic in **Figure 1b**. Allowing for two duplicate cases results in 25 distinct output levels. Thus, the technique achieves roughly $4^1/_2$ bits of resolution while using only three pins of a µC and five resistive elements. The transfer characteristic is symmetrical about midrail and does not extend to the supply rails, making it inherently suitable for use in

FIGURE 1



Providing three possible inputs—1, 0, and F (floating)—to the resistor network (a) produces a ternary-code transfer characteristic (b).

single-supply applications. Note that the characteristic is nonlinear, which shouldn't matter for many applications.

A practical circuit implementation uses a single-inline array to form the resistive network (**Figure 2**). The PIC16C84 (Microchip Technology, www.microchip.com) code in **Listing 1** uses a look-up table to convert binary inputs to the required ternary outputs. (You can download this **listing** and the related look-up table from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2189.)

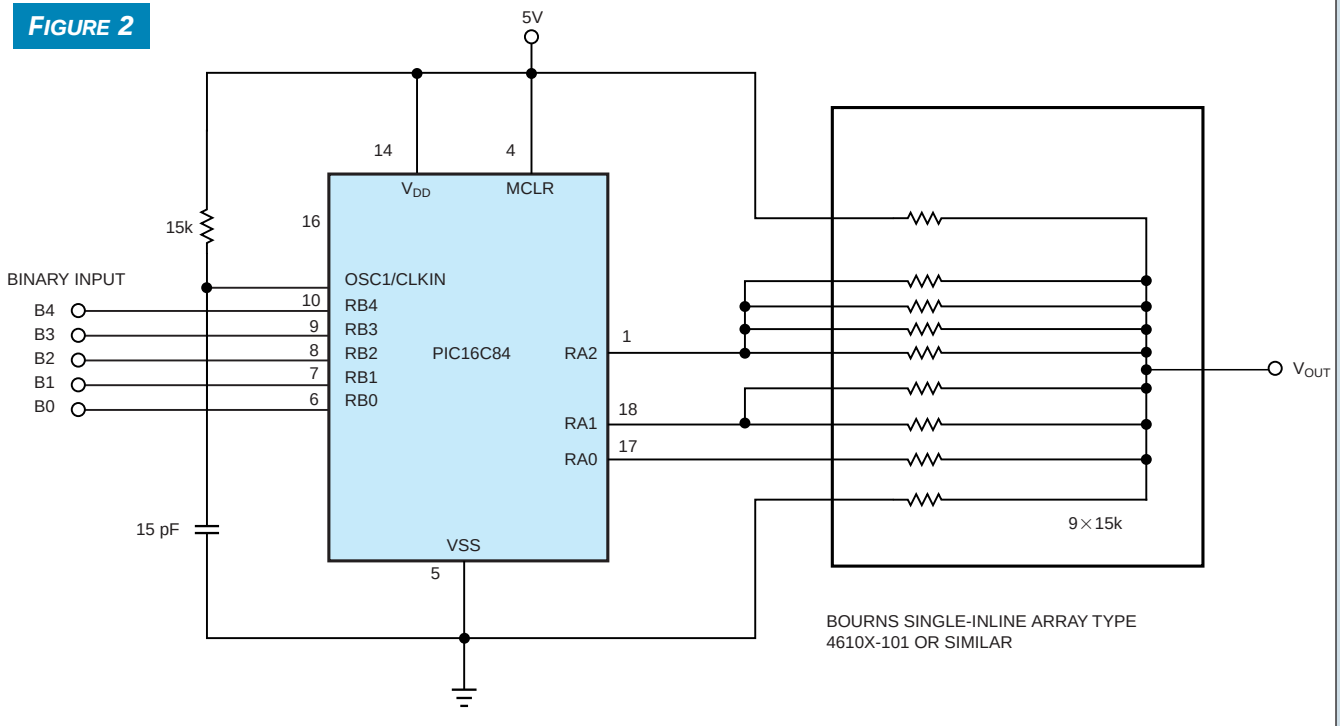
You can expand or contract this ternary technique within reason. For example, using four µC pins gives 75 distinct output levels, and even just two µC pins gives seven levels. By saving pins, the technique is ideally suited for use with the PIC12C50X family, which has very limited I/O in an eight-pin package. (DI #2189) EDN

To Vote For This Design, Circle No. 528

LISTING 1—PIC16C84 DAC CODE

```
; Title:    c:\edn\cheapdac\code_v1.asm
; Author:   Jes Dean
; Dated:    20/10/97
; Function: Program continually loops, reading 5 bit binary i/p
;           from RB4-0, and outputting ternary codes on RA2-0
;           suitable for driving resistor array as demonstrator.

        LIST    P=16C84

PC              equ     0x02 ; Relevant system registers...
STATUS          equ     0x03
PORTA           equ     0x05
PORTB           equ     0x06
TRISA           equ     0x85

temp            equ     0x10 ; Temporary storage.

        org     0x00
        goto    main

        org     0x10
; For brevity, no explicit initialisation done. PORTA, PORTB initialise
; as inputs with no weak pull-ups, and all interrupts are disabled.
main    movf    PORTB,0         ; Read PORTB
        movwf   temp            ; and store.
        sublw   26
        btfsc   STATUS,0        ; IF 0=<w=<26
        goto    inrange         ; THEN temp is in correct range.
        movlw   26              ; ELSE
        movwf   temp            ; Limit temp to 26.

inrange movf    temp,0          ; Copy temp to w
        call    ra_lut          ; and call look up table.
        bsf     STATUS,5        ; Enable addressing of TRISA (in Bank 1)
        movwf   TRISA           ; Set RA2-0 as inputs or outputs accordingly.
                                ;   (Note RA4,3 will be set spuriously.)
        bcf     STATUS,5        ; Disable addressing of TRISA (in Bank 1)
        movwf   temp
        swapf   temp,0          ;
        movwf   PORTA           ; Put relevant data on pins set as outputs.
                                ;   (Note RA4,3 may be set spuriously.)

        movlw   0xff            ; Delay loop to allow value to
        movwf   temp            ; settle for measurement purposes.
loop1   decfsz  temp,1
        goto    loop1

        goto    main            ; Repeat eternally.
```

FIGURE 2



A simple DAC demonstration circuit comprises a PIC16C84 µC and a single-inline-resistor array.

Undersampling extends utility of digital scopes

ROBERT J INKOL, DEFENCE RESEARCH ESTABLISHMENT, OTTAWA, ON, CANADA

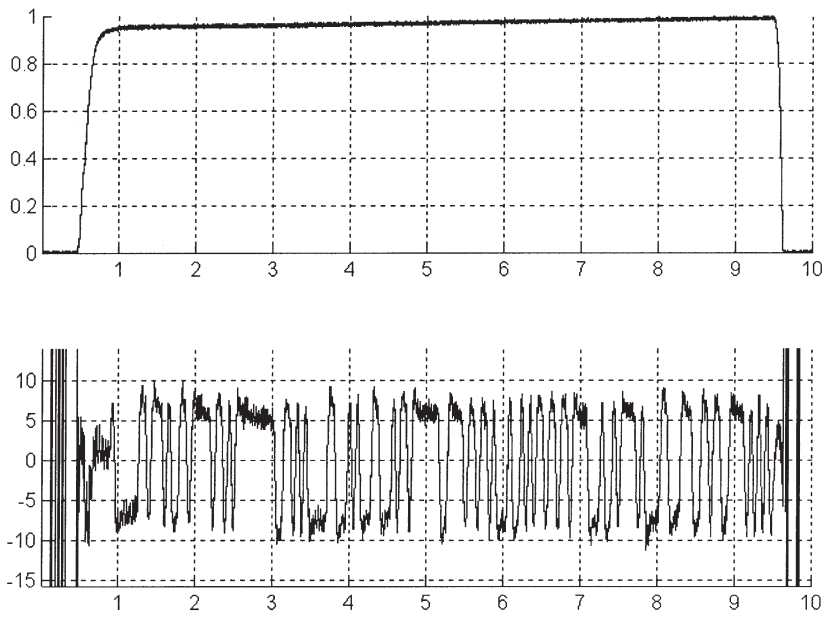
By undersampling the input signal, you can use a digital oscilloscope to digitize and display or collect signal data of even RF and IF signals. To sample a signal without aliasing, you must use a sampling rate that satisfies the relationship

$$f_M = f_S/2$$

where f_M is the maximum allowable signal frequency and f_S is the sampling rate. If you want the sampled signal data to directly form a high-quality visual representation of the signal waveform, a considerably higher sampling rate is necessary. A practical implication of this equation for high-frequency signals is that you can acquire sequences of signal data for only short periods before the available memory is filled.

However, many high-frequency signals, such as RF and IF signals in communications systems, are bandpass signals whose bandwidths are very small relative to the center frequencies. Consequently, you can intentionally undersample these signals so that their spectral components alias to lower frequencies. To ensure that the spectral

FIGURE 1



An undersampling technique helps capture the envelope (a) and instantaneous frequency (b) of a single hop from a typical frequency-hopping radio using FSK modulation.

$$f_L \geq mf_S,$$

components of the signal do not fold over on themselves or become reversed, you must choose the sampling rate, f_S,

$$f_U < (2m+1)f_S/2,$$

such that simultaneous solutions exist for
and
where $m=f_L/f_S$ (an integer) and f_L and f_U are the respective upper and lower bounds for the bandwidth that the signal of interest occupies. The undersampled signal differs from the original signal by a downward shift in frequency of mf_S. In carrying out this concept, you need to disable analog low-pass filters at the oscilloscope input or ensure that the filters have a cutoff frequency high enough to preserve the signal information.

You can apply this concept to practical applications, such as the digitization, storage, and analysis of the RF-signal output from a VHF radio that employs frequency-hopping techniques. In this application, the radio was programmed so that the signal would hop to a carrier frequency of 71.9 MHz at frequent intervals. A bandpass filter centered about 70 MHz with a bandwidth of 5.6 MHz attenuated and band-limited the transmitted signal. A LeCroy (www.lecroy.com) 9354 digital sampling oscilloscope, operating at a sampling rate of 10 MHz, generates records of 100k data samples. The choice of carrier frequency and sampling rate results in the aliasing of the RF signal to 1.9 MHz. Note that the use of a sampling rate of 200 or 250 MHz would not allow the storage of sufficient data for a complete hop.

You can then use Mathworks' (www.mathworks.com) Matlab to implement signal-processing algorithms for amplitude and frequency demodulation of data sequences associated with individual hops. **Figure 1** shows the time-domain behavior of the amplitude and instantaneous frequency measured for a representative hop signal. You can discern the FSK modulation in the switching of the instantaneous frequency between two discrete frequencies. (DI #2190) EDN

To Vote For This Design, Circle No. 529

Basic stamp computer eases prototyping hassles

MED DYER, JABRA CORP, SAN DIEGO, CA

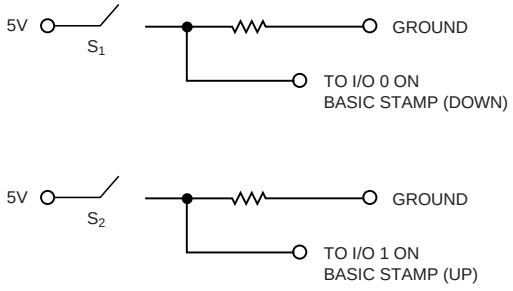
Many of today's complex ICs use a common three-wire serial interface to provide programming access. These ICs typically end up in circuits that include a μC or a CPU, and the controller or CPU then has the burden of programming the serially controlled parts. During early prototyping of a project, however, the μC or CPU may be unavailable for use: The hardware may be absent, or the complex programming necessary to fully use the CPU may not be in place.

For projects such as these—for which serial programmability is necessary but no resident host exists—you can use a simple and cost-effective prototyping tool: the Basic Stamp II (BSII). A battery-powered BSII is a unique tool for any engineer who regularly works on development projects and with many applications in the lab. With just a few pushbuttons and some minimal interface to the project at hand, powerful portable control is possible—with a minimal learning curve and modest expenditure.

The BSII from Parallax Inc (Melville, NY) is a complete Basic-programmable computer, all contained on a 24-pin DIP module. When plugged into its optional carrier, the BSII offers a serial port for connection to a PC, a battery clip for 9V battery power, a reset button, and plenty of room for the minimal additional circuitry needed for most prototyping applications. The BSII includes a simple PC-based Basic compiler for developing BSII code, which you download to the BSII's EEPROM for execution. Although a quick read of the BSII manual brings to mind a host of useful applications for the typical R&D engineer, one application is particularly handy: using the BSII to program three-wire serial devices.

One example of a three-wire serial device is the LM1973 three-channel audio attenuator from National Semiconductor (Santa Clara, CA). This part can serve as a three-channel audio pot, and a three-wire serial interface performs channel

FIGURE 1



Two normally open pushbuttons—one for "up" and one for "down"—provide an active high to the Basic stamp when depressed.

LISTING 1—BASIC STAMP II SERIAL-PROGRAMMING CODE

```
'  This code uses the BASIC Stamp II to provide serial programming
'  data to a digital potentiometer IC using the standard three-wire
'  serial interface. Two buttons are added to the standard BSII board.
'  These buttons serve as Up/Down pushbuttons and are active high
'
'
'              /        10k
'    +5v ----/  --+--/\/\/\---Gnd
'           SW1  |
'                +-----------I/O 0 on BASIC Stamp (down)
'
'
'              /        10k
'    +5v ----/  --+--/\/\/\---Gnd
'           SW2  |
'                +-----------I/O 1 on BASIC Stamp (up)
'
'
'
'  The following three BSII outputs should be connected to the digital pot:
'  I/O 8  --  DATA on digital pot  (pin 11 for LM1973)
'  I/O 9  --  CLOCK on digital pot  (pin 9 for LM1973)
'  I/O 10 --  LOAD on digital pot  (pin 10 for LM1973)


DataPin           con   8
ClockPin    con   9
LoadPin           con   10

'Define workspace for BUTTON variables
Up              var     byte
Down            var     byte

'Define storage for current pot data setting
Data16            var   word


'Initialize workspace for BUTTON variables
Up = 0
Down = 0

'Initialize digital pot to zero
high        LoadPin
Data16 = %0000000000000000
pause 1500
low         LoadPin
shiftout    DataPin, ClockPin, 1, [Data16\16]
high        LoadPin


'Loop until a button (up or down) is pressed

BLoop:

button 1,1,200,225,Up,1,MoveUp

button 0,1,200,225,Down,1,MoveDown

pause 500
goto BLoop


'The "Up" button was pressed, increase pot setting by 1 unless maxed already
MoveUp:
if Data16 & %11111111 => %11111111 then BLoop
Data16 = Data16 + 1

low LoadPin
shiftout DataPin, ClockPin, 1, [Data16\16]
high  LoadPin

goto  BLoop

'The "Down" button was pressed, decrease pot setting by 1 unless zero already
MoveDown:

if Data16 & %11111111 < 00000001 then BLoop
Data16 = Data16 - 1

low LoadPin
shiftout DataPin, ClockPin, 1, [Data16\16]
high  LoadPin

goto  BLoop
```

selection and gain control. The format is simple: The system must first pull the Load line on the LM1973 low and then pulse 16 clocks on the Clock line with a corresponding 16 bits of input data on the Data-in line. The 16 bits of data comprise 8 address bits, which select one of the three channels and 8 attenuation bits, which select the attenuation setting for the selected channel. This data format is most significant bit first with the address bits first and the attenuation bits immediately after. This straightforward format could be difficult to implement in a prototype without a controller, but the BSII makes a "kluge" programmer devilishly simple.

This example uses only one channel of the LM1973, although using all three channels is just as simple. Two normally open pushbuttons—one for up and one for down—provide an active high when depressed (**Figure 1**). You also need to connect the following three BSII outputs to the digital potentiometer: I/O 8 to Data (Pin 11 for LM1973), I/O 9 to Clock (Pin 9 of LM1973), and I/O 10 to Load (Pin 10 of LM1973)

The meat of the code required for this application comprises two BSII instructions: Button and Shiftout. Button checks the status of a BSII input line and allows for branching according to that status. This example monitors the up and down buttons until one is pressed and then reprograms the LM1973 accordingly. The Shiftout instruction shifts out a data word synchronously along with a clock output. You select the appropriate output pins and specify a most-significant-bit-first protocol, as the LM1973 demands.

This minimal code provides up/down programming on a single channel but is easily expandable to provide other functions, including data readback from the serial device, multiple channels, or custom programming sequences. You can easily edit the code on the PC, debug the code with the BSII connected to the PC, and then download the code and carry it away on the BSII.

Listing 1 is available for downloading from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go to the Software Center to download the file from DI-SIG #2178. (DI #2178) EDN

To Vote For This Design, Circle No. 530

10-kHz VFC uses charge-pump variation

STEPHEN WOODWARD, UNIVERSITY OF NORTH CAROLINA, CHAPEL HILL, NC

A diode-capacitor charge pump is the starting point for many voltage-to-frequency-converter (VFC) designs. The circuit in **Figure 1** uses a variation on that classic theme to achieve linearity less than 0.05%, micropower operation of approximately 10-µA total draw from a 5 to 36V rail, and bipolar-input capability. The basis for these features is the switchable-polarity, self-compensating charge pump comprising D_1 to D_4, C_1 to C_4, and CMOS switches S_2 and S_3. Although simple in concept, VFCs using diode-capacitor pumps suffer from the need to cope with the nonideal characteristics of diodes used as analog switches.

Temperature-dependent forward-voltage drop, junction and stray capacitance, and reverse leakage current all conspire to limit converter accuracy. The stray capacitance and leakage current are especially troublesome in low-power applications, in which the need to minimize pump-current consumption limits the size of the pump capacitors. Because the total amount of charge pumped in each converter cycle is minimal, the error sources are proportionally more significant and thus harder to control and compensate. The unique pump circuit in this converter comprises two distinct halves: D_1, D_2, C_1, and C_2 generate a frequency-proportional current that closes the VFC's feedback loop, and D_3, D_4, C_3, and C_4 generate an error-correcting compensation current.

If you assume that C_2=C_2=C_3=C_4 and equality of diode forward drops (V_D) and stray capacitance (C_S), then the net feedback current from the pump is

$$f_{OUT}(2C_1+C_S)\left[4.55V\times\frac{2C_1}{2C_1+C_S}-2V_D-(4.55V\times\frac{C_1}{2C_1+C_S}-2V_D)\right]$$
$$=f_{OUT}(2C_1+C_S)\left[4.55V\times\frac{2C_1-C_1}{2C_1+C_S}+2V_D-2V_D\right]$$
$$=f_{OUT}\times4.55\times C_1=f_{OUT}\times10^{-4}\,\mu A/Hz.$$

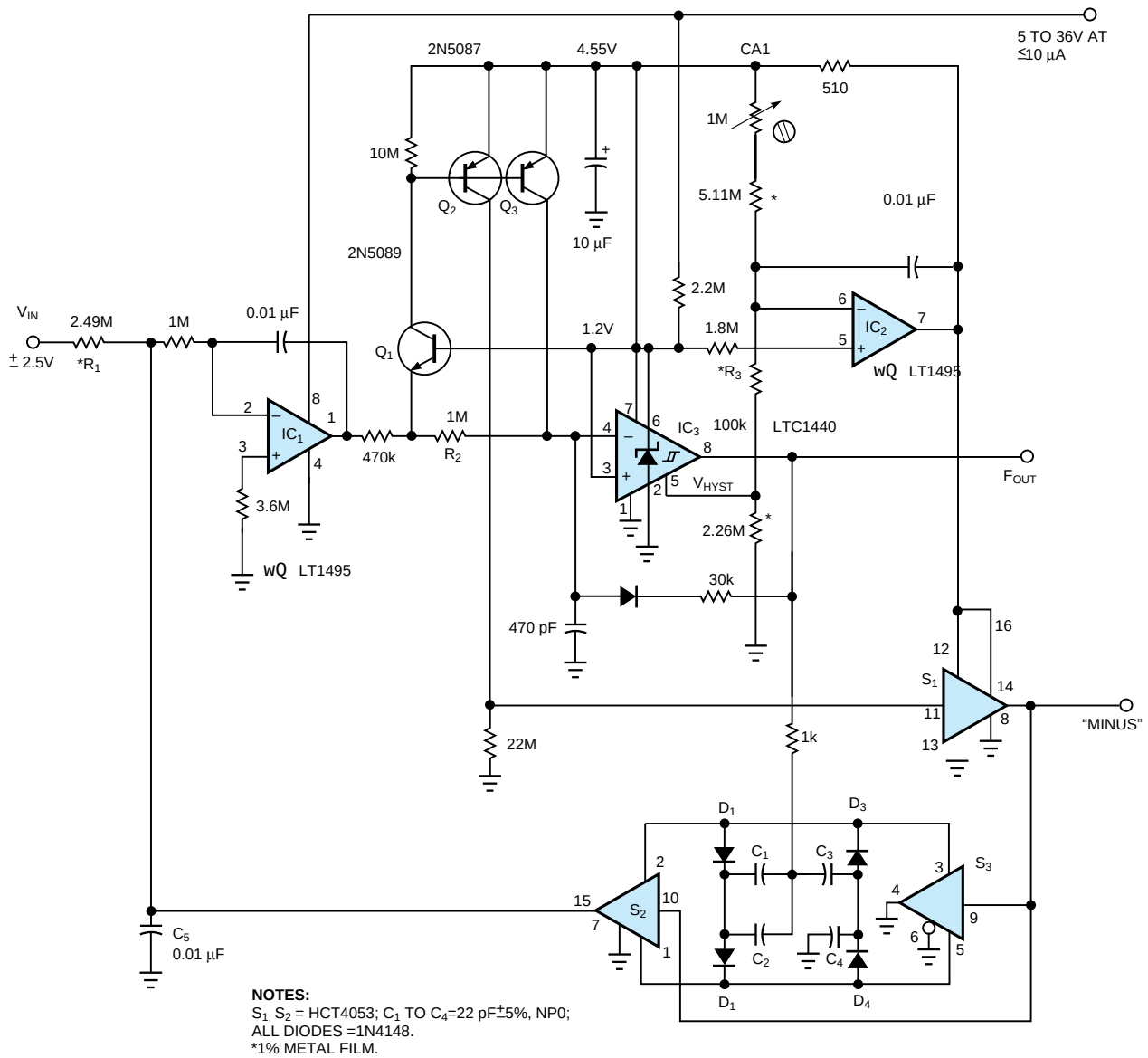
You not only obtain compensation for the bothersome V_Ds, but also eliminate the effects of stray capacitance in the bargain. Operation of the converter depends on integrator IC_1's control of multivibrator IC_3. The combination is such that f_{OUT}=0 when IC_1's output is 1.2V. If, for example, V_{IN}>0V, IC_1 ramps negative. As IC_1 ramps through approximately 0.8V, Q_1 begins to conduct, thereby turning on both Q_2 and Q_3. Q_2 drives S_1 to the "plus" polarity state, providing a status signal to the connected system (typically, a gated up/down counter). The status signal indicates the presence of a positive V_{IN}. S_1 sets up S_2 and S_3 to provide a negative feedback current to C_5. Subsequently, Q_3's collector current causes IC_3's f_{OUT} to increase until 1E–7×f_{OUT}=V_{IN}/R_1=4 kHz/V

for the values shown, and the integrator is thus balanced.

V_{IN}<0V causes IC_1 to ramp positive, turning off the Q_1-Q_2-Q_3 transistor trio. This action causes S_1 to generate a "minus" status and set up S_2 and S_3 to generate a positive feedback current. The loop adjusts f_{OUT} until $1E{-}7{=}{-}V_{IN}/R_1$, as in the case of V_{IN}>0V. The converter's overall temperature coefficient depends on matching all pump capacitances, including the pc-board contribution to C_S parasitics. A ±5% capacitance tolerance is good enough to reduce the charge-pump temperature coefficient to approximately 50 ppm/°C. The converter linearity is ±0.03%, and the current draw is an unexcelled 6.5 to 10 µA as f_{OUT} goes from 0 to 10 kHz. IC_1's approximately 300-µV input-offset spec determines the converter's zero offset. IC_2's regulation of the 4.55V reference affords good power-supply rejection, yielding undiminished accuracy for supply voltages from 5 to 36V. (DI #2183) EDN

To Vote For This Design, Circle No. 531

FIGURE 1



A fleapower bipolar-input VFC uses a modified charge-pump technique to achieve high linearity with power supplies ranging from 5 to 36V.

Single IC biases LCD and GaAsFET amplifier

JOHN WETTROTH, MAXIM INTEGRATED PRODUCTS, APEX, NC

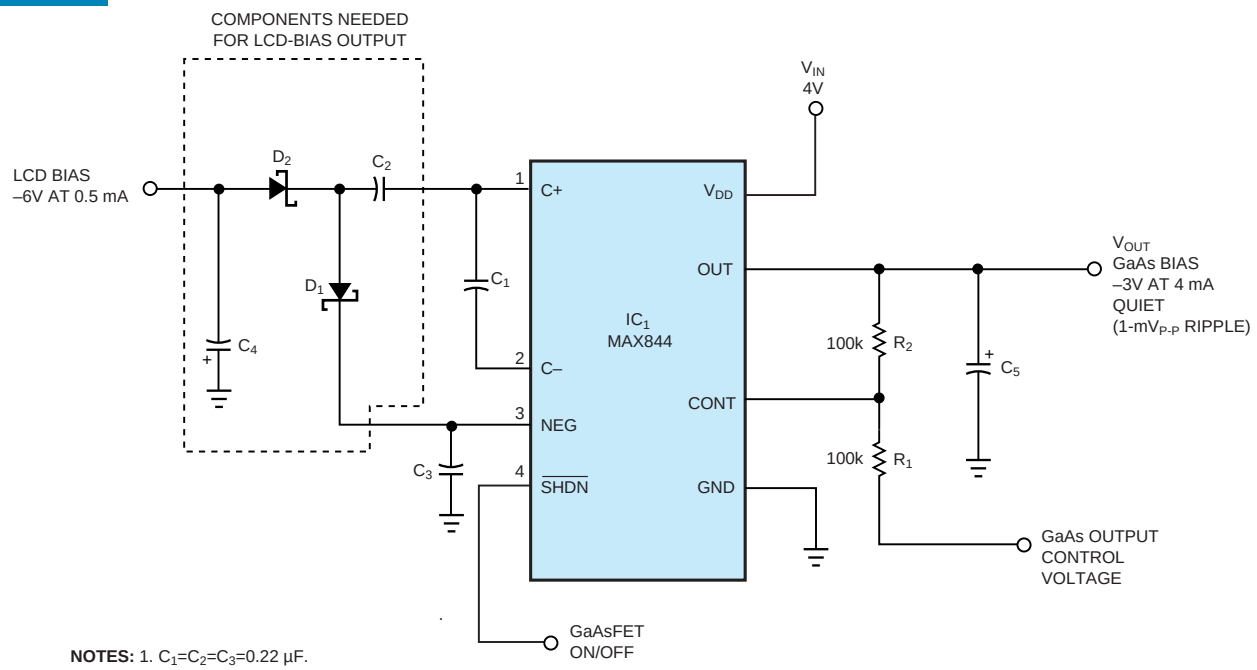
Operating from a lithium-ion cell or a four-cell NiCd battery, the circuit in **Figure 1** provides –6V bias for LCDs and a separate, quiet negative bias for a GaAsFET power amplifier. This bias combination exists in cell phones, two-way pagers, wireless modems, and many other wireless devices. Not long ago, when most GaAsFET amplifiers required a positive voltage of 6V or more, you could easily obtain the LCD bias by simply inverting the power-amplifier voltage. With the advent of lower voltage power amplifiers and single Li-ion supplies, a negative doubler became necessary for the LCD bias. The various bias-generation techniques in use are generally bulky and require multiple ICs. One alternative is to select an LCD with lower negative bias voltage, but that approach compromises the LCD's temperature performance, contrast, and cost in favor of a simpler bias supply.

For most systems, the approach is to use two ICs: a negative doubling inverter, such as the MAX865, which provides a negative LCD bias of approximately –6V, and a linear regulator to provide the –3V GaAsFET bias. However, even two ICs can pose a problem in tiny systems. Moreover, a simple linear regulator may generate too much noise, and noise in the GaAsFET bias can appear in the transmitted RF signal. IC_1, which includes a charge-pump inverter and a low-noise linear regulator in an SO-8 package, generates a quiet GaAsFET bias by design. It operates from supplies as low as 2.5V and produces a negative bias voltage with only 1 mV p-p ripple. You can change the bias level by adjusting R_1 and R_2, according to instructions in the data sheet.

Circuitry in the dashed line provides the –6V LCD bias. A square-wave signal from the charge pump (Pin 1) adds to the unregulated negative voltage at Pin 3 to form a negative, doubled version of the input voltage. The voltage loss (two diode drops) is minimal because of the LCD's low bias current and the use of low-drop Schottky diodes. The diodes drop approximately 0.2V; a Li-ion cell can thus produce an LCD bias greater than –6V. (DI #2179) EDN

To Vote For This Design, Circle No. 532

FIGURE 1



NOTES: 1. $C_1=C_2=C_3=0.22\ \mu F$.
2. $C_4=C_5=4.7\ \mu F$.
3. $D_1=D_2=SGL41$.
4. $V_{OUT}=(CONTROL\ VOLTAGE)\times\frac{R_2}{R_1}$
5. IF POWER-DOWN IS REQUIRED, USE MAX843 FOR IC_1.

A charge-pump/linear-regulator IC produces two negative bias voltages for wireless applications.

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____ Fax _____

E-mail _____

My e-mail address may be published Yes____ No____

Company _____

Address _____

_____

_____

Country _____ ZIP _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

The winning Design Idea for the Sept 12, 1997, issue is entitled "Solar cells implement low-cost illuminator," submitted by Dennis Eichenberg (Parma Heights, OH).

Circuit protects FPGAs from killer spikes

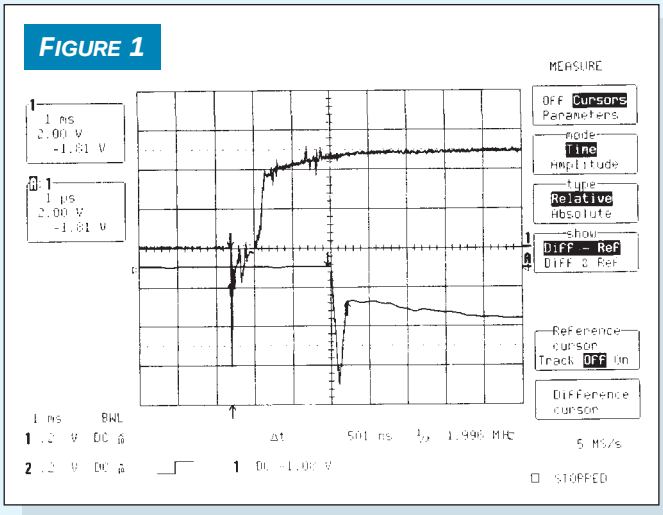
NELSON NGUYEN, ANRITSU CORP, MORGAN HILL, CA

A project using Xilinx FPGAs brought an interesting problem to light. When you turn on the board, one FPGA in three succumbs to this problem. A lot of frustration and testing uncovered a negative-going spike (**Figure 1**) in the 5V line from the dc/dc converter. The system uses a dc/dc converter to convert –48V to 5V and other voltages. The spike occurs before the converter delivers its intended 5V. Spikes greater than 5V would kill the FPGA with the shortest path to the converter. The circuit in **Figure 2** solves the problem.

Because the spike occurs before the 5V supply line turns on, to prevent the spike from destroying the FPGA, you should open the 5V path when you turn on the power switch and then close the path when the 5V supply voltage is present. The R_{19}-C_{13} RC network provides a delay in turning relay K_1 on. The turn-on voltage for K_1 is approximately 3.7V. The voltage divider comprising R_{19} and K_1's coil resistance (approximately 780Ω for an NEC EA2-5NU) provides a voltage at the junction of C_{13} and R_{19} sufficient to turn K_1 on. The value of C_{13} sets the delay at approximately 2 msec. (DI #2181) EDN

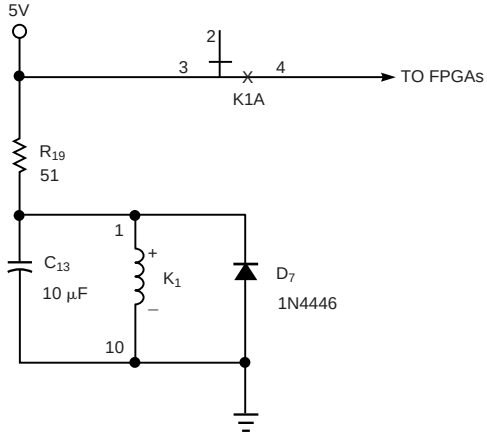
To Vote For This Design, Circle No. 366

FIGURE 1



A large negative spike (top waveform) in the turn-on waveform of the 5V supply line is an effective FPGA destroyer.

FIGURE 2



NOTE: K_1 = NEC EA2 – 5NU.

A simple RC network and a relay provide a 2-msec turn-on delay in the power-supply line to the FPGAs, thereby blocking the killer negative spike.

Bootstrapped boost converter operates at 1.8V

TOM GROSS, LINEAR TECHNOLOGY, MILPITAS, CA

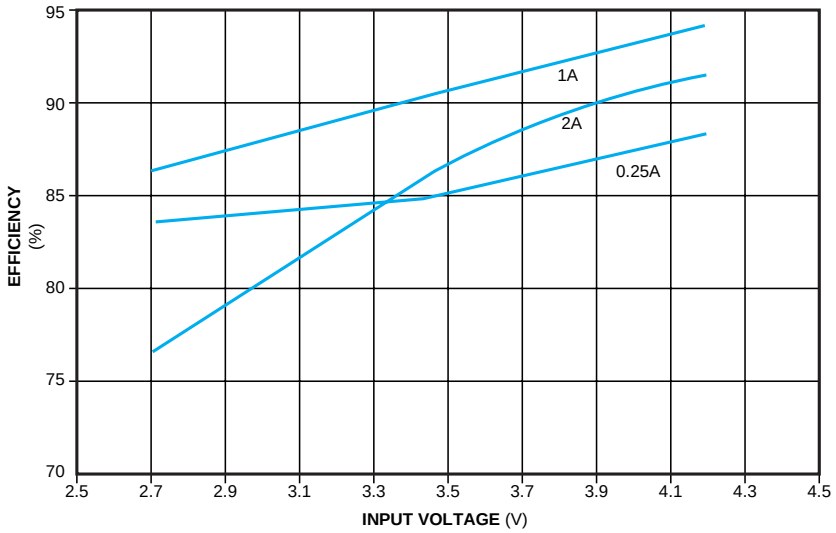
Many circuits, such as those that use batteries or solar cells, must operate in the face of decreasing supply voltages. The circuit in **Figure 1** maintains the maximum load current as the supply voltage drops. The regulator boosts a 2.5 to 4.2V input to 5V and provides 2A load current, for 10W of output power. The circuit is a bootstrapped synchronous boost regulator that uses an LTC1266 synchronous-regulator controller. Diodes D_1 through D_5 allow the circuit to start up using the low input voltage and then to receive its power from the higher output voltage during normal operation. The crucial elements in the circuit are the switches: two IRF7401 n-channel MOSFETs. The MOSFETs receive full enhancement at low gate-source voltages. (At V_{GS}=2V, the peak drain current is 15A.) The low enhancement voltages allow the circuit to start with low input voltages.

This low-voltage capability is important for low-series-cell-count, battery-powered systems. Diodes D_3 and D_4, along with capacitor C_2, form a charge-pump circuit, which the controller uses for the MOSFETs' gate drive. Because the circuit receives its power from the 5V output voltage, the cir-

cuit still operates if the input supply voltage drops below the IC's minimum input voltage. This bootstrapping allows the circuit to start up when the input voltage is below the IC's 3.5V minimum input spec. With a 1A load, the regulator operates with inputs as low as 1.8V. **Figure 2** shows the regulator's efficiency vs the input voltage with three load currents. With 2A load current, the efficiency drops as the input voltage decreases, because of the higher power losses in the inductor. A larger inductor would provide increased efficiency or allow for greater load currents. (DI #2185) EDN

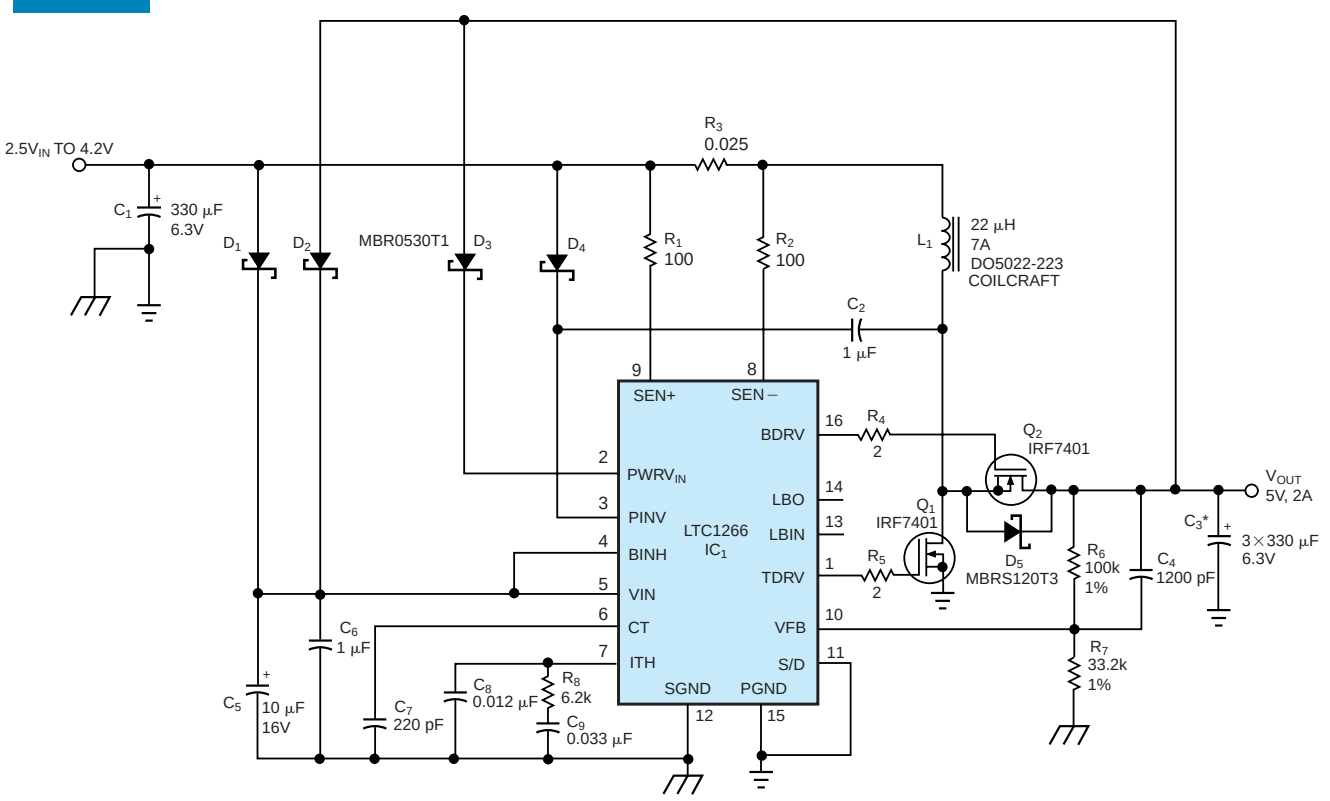
To Vote For This Design, Circle No. 367

FIGURE 2



The conversion efficiency for the boost regulator in Figure 1 averages 87% overall. The circuit provides its best efficiency for all input voltages with a 1A load.

FIGURE 1



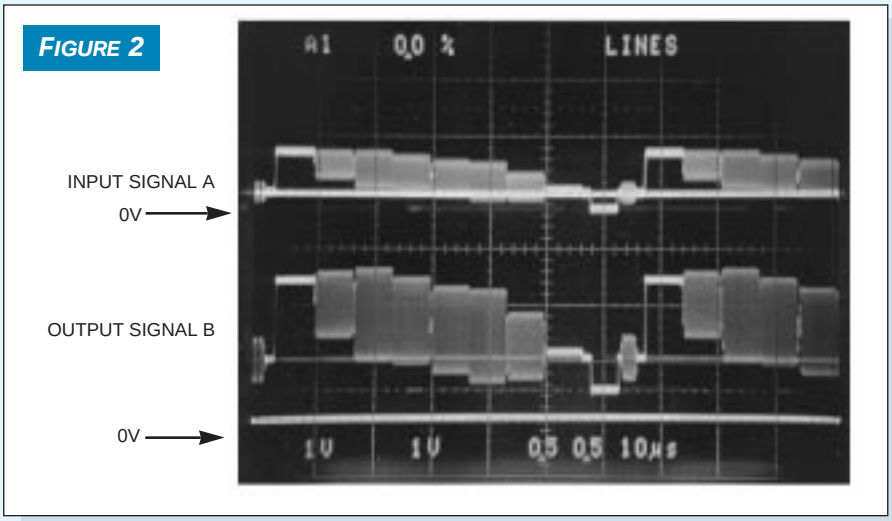
A charge-pump arrangement allows this boost regulator to operate with input voltages far below the minimum specified value for the regulator IC.

Video circuit clamps under all conditions

NICHOLAS GRAY AND TERRANCE SMITH, NATIONAL SEMICONDUCTOR, SANTA CLARA, CA

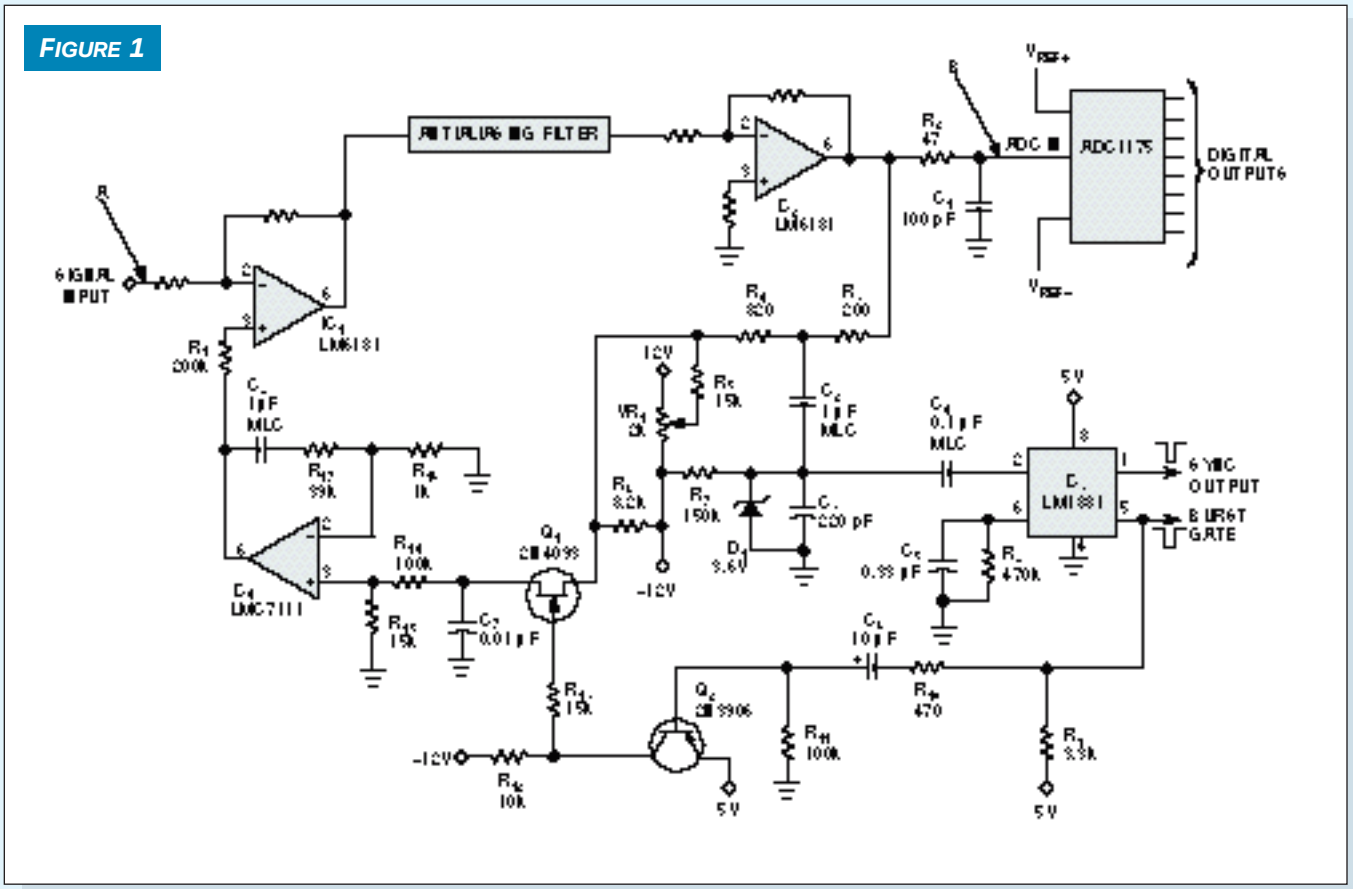
Many video-circuit clamps operate well in the presence of a composite-video signal but cannot achieve a clamp level with signals other than composite video or in the absence of an input signal. The circuit in **Figure 1**, developed for the ADC1175 (a popular and inexpensive, high-performance, 8-bit, 20M-sample/sec ADC), provides the normal back-porch clamp function to the input of the ADC in the presence of a composite-video signal. The circuit further ensures that the voltage presented to the ADC is within its correct operating

FIGURE 2



This scope photo shows the offset at Point B relative to Point A in Figure 1. Note also the gain the circuit provides.

FIGURE 1



The clamp level at Point B in this circuit is independent of the average level at Point A. The circuit thus keeps signal levels with the ADC's input range.

range in the absence of an input signal and forces any signal other than composite video to be within the ADC's input common-mode range.

The circuit accomplishes video clamping by building a control loop that forces the dc voltage at IC_2's output to a desired level during the blanking period. This level, approximately 25% of full scale for a composite-video signal, forces the ADC's output-pedestal (blanking) level to an 8-bit code of approximately 64. The simple filter comprising R_3 and C_3 bandlimits the signal at the output of IC_2. This high-frequency attenuation is necessary to prevent noise spikes from upsetting the operation of the LM1881. The LM1881 is a video sync-separator chip that produces burst-gate pulses at its Pin 5 when a composite-video signal is present at Pin 2.

The burst-gate output of the LM1881 serves to sample the blanking level of the video signal. Potentiometer VR_1 and R_5 produce an adjustable offset in the signal path when Q_1 gates on. During the blanking period, the ac-coupled burst-gate signal pulls Q_2's base low (to approximately 4V), thus pulling Q_1's gate high, thereby sampling and storing the sum of the video-blanking level and the dc offset from VR_1 onto C_7. At times other than the back-porch interval, Pin 5 of the LM1881 is high, and Q_2 is off, thereby turning Q_1 off. Divider R_{14}-R_{15} attenuates the voltage on C_7 to ensure sufficient phase margin in the clamp loop. IC_4 is an integrator that averages the attenuated dc value over many samples. This average sums with the input signal in IC_1.

If the integration time is too small, the result could be shading across the display. A long integration results in slow, perceptible adjustments when switching between fields with large differences in average brightness. The dc feedback path for IC_4 is through IC_1 and IC_2. If no video signal exists or if the input signal has no sync, R_{11} holds Q_2 on, thus holding the video output of the circuit within the ADC's operating range. With VR_1 centered, the level halfway between the positive and negative peaks of the input signal clamps at approximately 1.6V, or approximately halfway between the high and low reference voltages (2.6 and 0.6V, respectively) of the ADC1175. The circuit achieves an effective number of bits of 7.5, corresponding to a signal-to-noise and distortion of 47 dB. **Figure 2** shows the offset at Point B in **Figure 1**, relative to the voltage at Point A. (DI #2184) EDN

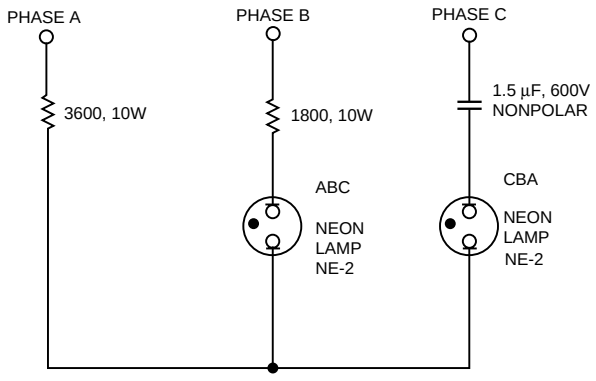
To Vote For This Design, Circle No. 368

$5 junk-box circuit determines phase sequence

HUGH ADAMS, FORT WALTON BEACH, FL

Have you ever wondered which way a blower motor is going to turn when you plug it into another socket, or have you ever inherited the task of modifying three-phase wiring in your plant? The circuit in **Figure 1** is a simple, approximately $5 phase sequencer that you can probably build from parts in your junk box and save approximately $50 to boot. The component values reflect 60-Hz operation, but the design equations in **Figures 2** and **3** allow you to select values for other frequencies. The equations are in MathCAD spread-

FIGURE 1



The brighter of the two neon lamps indicates the phase sequence, either ABC or CBA.

FIGURE 2

$$f=60 \quad C=1.5\cdot10^{-6} \quad V=0 \quad R1=3600 \quad R2=1800$$

$$x=\frac{1}{2\cdot\pi\cdot f\cdot C} \qquad x=1.768\cdot10^{3}$$

$$IA=\frac{1}{R1}\cdot(120\cdot\cos(2\cdot\pi\cdot f)-V)$$

$$IB=\frac{1}{R2}\left(120\cdot\cos\left(2\cdot\pi\cdot f-2\cdot\frac{\pi}{3}\right)-V-90\right)$$

$$IC=\frac{1}{\left[\left(\frac{1}{2\cdot\pi\cdot f\cdot C}\right)^{2}\right]^{wQ}}\cdot\left(120\cdot\cos\left(2\cdot\pi\cdot f-4\cdot\frac{\pi}{3}-\frac{\pi}{4}\right)-V-90\right)$$

$$IA=0.033 \quad IB=-0.083 \quad IC=-0.033$$

MathCAD spreadsheet equations show a higher current in Phase B than in Phase C of the circuit in Figure 1; therefore, the phase sequence is ABC.

sheet format, but almost any other spreadsheet would do.

Referring to **Figure 1** and the equations, you can see that the neon bulb that glows brighter indicates the phase sequence, or phase-rotation order, ABC or CBA. The bulb glows brighter because it carries more current because of the phase shift the 1.5-μF capacitor provides. You can verify this assertion by examining the two sets of equations. Note that the two sets of equations have different expressions for I_B and I_C. In one, I_B lags I_A by $2\pi/3$; in the other, it lags by $4\pi/3$, and vice versa for I_C. The equations provide the mathematical way of reversing the phase sequence, and, as you can see, the two currents I_B and I_C reverse their relative magnitudes as the phase rotation reverses. (DI #2180) EDN

To Vote For This Design, Circle No. 369

FIGURE 3

$f=60$
$C=1.5\cdot10^{-6}$
$V=0$
$R1=3600 \quad R2=1800$

$x=\frac{1}{2\cdot\pi\cdot f\cdot C}$

$x=1.768\cdot10^{3}$

$IA=\frac{1}{R1}\cdot(120\cdot\cos(2\cdot\pi\cdot f)-V)$

$IB=\frac{1}{R2}\cdot\left(120\cdot\cos\left(2\cdot\pi\cdot f-4\cdot\frac{\pi}{3}\right)-V-90\right)$

$IC=\frac{1}{\left[\left(\frac{1}{2\cdot\pi\cdot f\cdot C}\right)^2\right]^{wQ}}\cdot\left(120\cdot\cos\left(2\cdot\pi\cdot f-2\cdot\left(\frac{\pi}{3}-\frac{\pi}{4}\right)\right)-V-90\right)$

$IA=0.033$
$IB=-0.083$
$IC=-0.116$

A CBA phase sequence produces a higher current, thus a brighter neon lamp, in Phase C of the circuit in Figure 1.

Piezo device generates buzz, beep, or chime

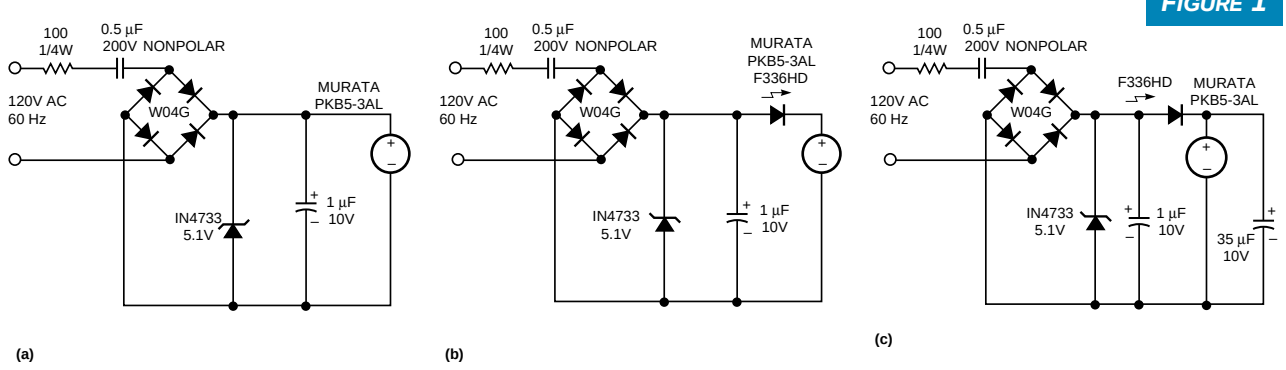
DENNIS EICHENBERG, PARMA HEIGHTS, OH

Piezoelectric buzzers, such as the Murata (Smyrna, GA) PKB5-3A in **Figure 1**, make excellent alarms. They're compact, lightweight, efficient, and reliable. However, a piezo alarm is a dc device; it requires additional circuitry to operate from an ac source. The circuits in **Figure 1** provide a simple and inexpensive way to obtain the dc drive. The W04G full-wave bridge rectifier produces a full-wave dc waveform from the 120V ac line. The 100Ω resistor protects the circuit from surges when you first apply power. The 5.5V 1N4733 zener diode protects the buzzer against high-voltage excursions. The 1-μF capacitor provides filtering for the buzzer.

The circuit in **Figure 1a** produces a true buzzer sound. The addition of an F336HD flashing LED (part number 276-036 at Radio Shack) in **Figure 1b** changes the alarm to a beeper, and it also provides a visual alarm. The LED produces a constant pulse of light at approximately 1 Hz without the addition of a time-constant capacitor. The LED starts immediately when you apply power, and it's insensitive to temperature variations. The addition of a 35-μF capacitor in parallel with the buzzer (**Figure 1c**) changes the audible alarm to a pleasing chime. The value of the capacitor is not critical; you can obtain various sound effects by varying it. (DI #2194) EDN

To Vote For This Design, Circle No. 370

FIGURE 1



A handful of inexpensive components configures a piezo alarm device as a buzzer (a), a beeper (b), or a chime (c).

Smart switch cuts transformer turn-on current

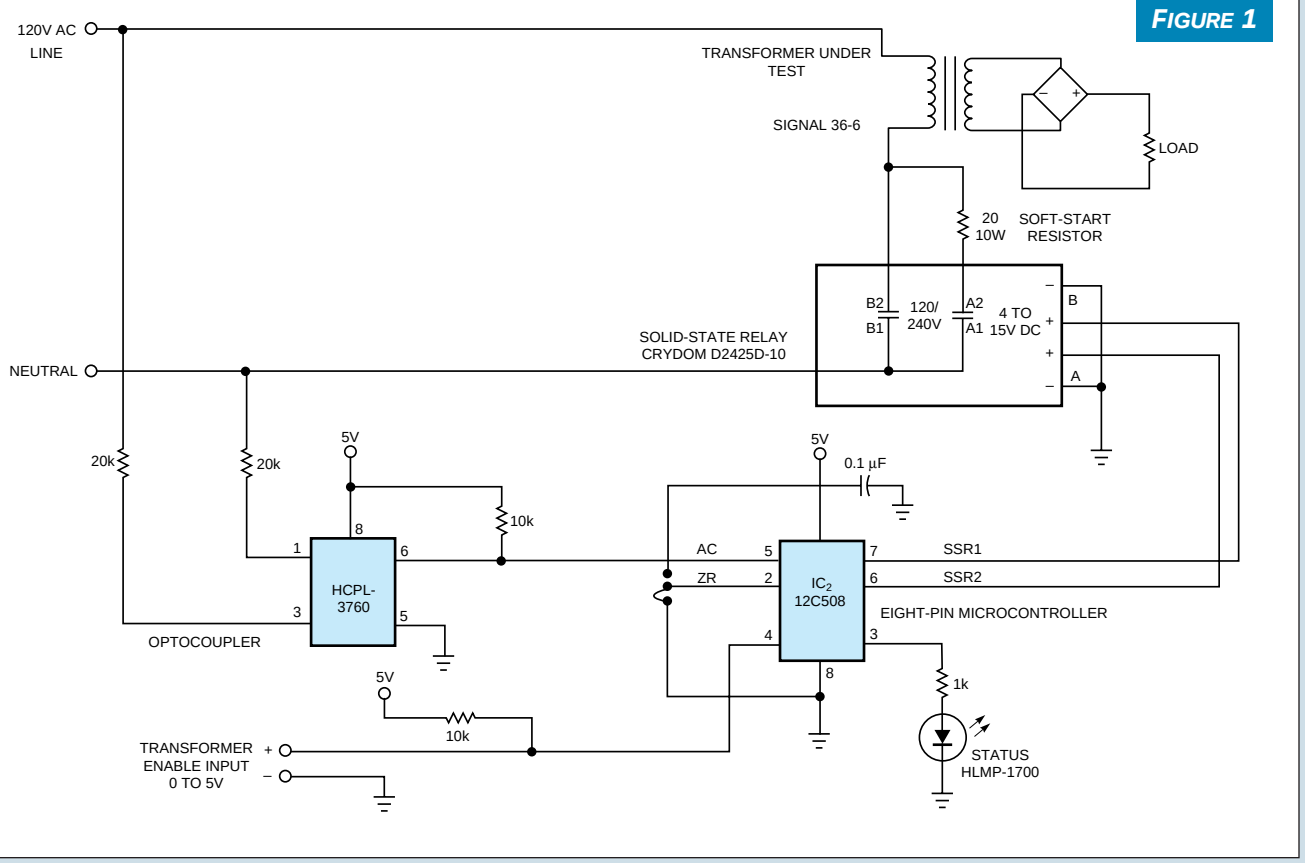
ROBERT LINDSEY, HANSVEDT EDM, URBANA, IL

Transformer-core saturation can cause inexplicable fuse blowing, system crashes, or premature switch and relay failure. When a core saturates, it loses its inductive characteristics; primary winding current can then reach extremely high values for several ac cycles. Turning on a transformer may seem fundamental, but in some power-supply designs and control applications, it can be a game of Russian roulette. Because transformers remain polarized when turned off, saturation occurrence is a function of the polarity and phase angle of the ac cycle when you switch the circuit on and off. The smart-switch circuit in **Figure 1** eliminates saturation, improves relay reliability, and provides a tool for determining transformer and relay performance.

The circuit goes beyond typical configurations using zero-crossing or peak-switching relays, by using the polarity of the ac cycle, known phase angles, and soft-starting techniques. **Figure 2** shows that the primary turn-on current of a 220-VA transformer can be disastrous when you use a zero-crossing relay. Trace R1 shows 46A peak with a saturated core. Trace 1 shows only a few amps with use of the smart-switch circuit. This large difference in current demonstrates the value of the smart switch in controlling transformer magnetization. Switching on during a positive half cycle and off during a negative half cycle or vice versa prevents most core saturation.

Peak switching of the ac voltage during turn-on and -off further reduces the susceptibility to core saturation, regardless of ac polarity. This reduction is an important consideration in the event of an uncontrolled power outage. **Figure 3**, trace R, shows the primary current with peak and same-polarity switching. The vertical scale in **Figures 2** and **3** is 10A per division, and Trace 2 is the relay control voltage. The primary current in **Figure 3** causes some core saturation (note that the current is not bipolar), but the saturation is much lower than that in **Figure 2**. Trace 1 shows the reduced primary current with the use of peak and opposite-polarity switching. Note that transformer designs vary widely; some may favor particular phase angles.

FIGURE 1



A µC-controlled smart switch prevents transformer-core saturation, thus averting system crashes and prolonging the life of power-supply relays.

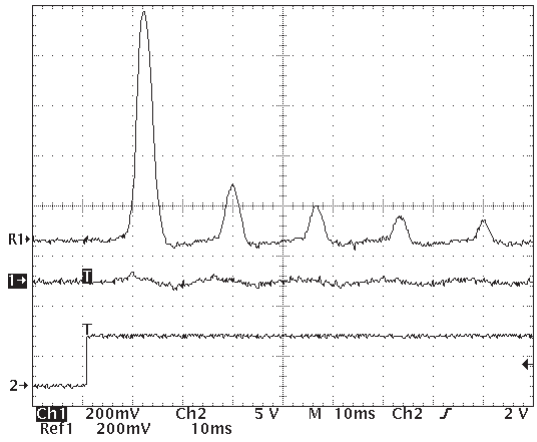
Inrush current from power-supply filter capacitors is also an important design consideration. By using a resistor, an inrush device, or an inductive input filter in the secondary winding, you can reduce this inrush surge. Another solution is to soft-start the transformer by using a resistor in the primary to limit inrush and saturation currents to an acceptable level. After a brief delay, a second solid-state relay shunts the resistor. The Microchip 12C508 µC uses its internal 4-MHz RC oscillator for all timing. The chip is simple, inexpensive, reliable, and well-suited for this application. For wide temperature variations, you can obtain more accurate timing by using a 32-kHz crystal. You can download **Listing 1**, the source code for the µC's operation, from *EDN*'s Web site www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2170.

You can use either zero-crossing or random relays, but the random type works better for transformers. Set Pin 4 high for zero-crossing relays and low for random-turn-on relays. The HCPL-3760 optocoupler determines the polarity and phase of the ac line. The coupler is configured as a near-zero detector. Its output is set to switch on at 50V ac and off at 25V ac.

LISTING 1—SOURCE CODE FOR TRANSFORMER SOLID-STATE-RELAY CONTROLLER

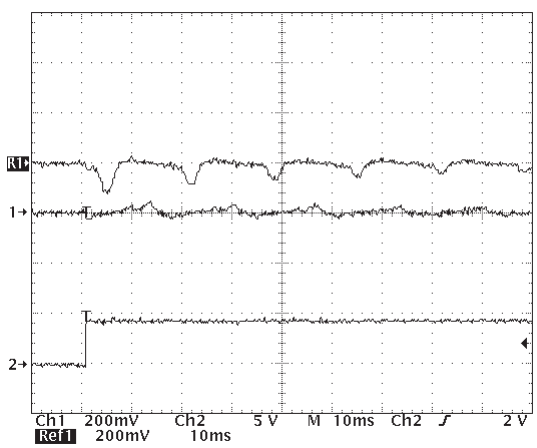
```
;
; ROBERT LINDSEY
; 1908 KENNY AVE, CHAMPAIGN, IL  61821
; 217 384-5900

; This is a smart switch for controlling a solid state relay that is used
; for controlling the AC primary power to a large transformer.  Transformers
; are noted for having extremely high inrush currents due to momentary
; core saturation if the polarization is correct. This version has an optional
; soft start switch that uses a resistor in the primary to limit current
; from the power supply filter capacitors and core saturation. The primary
; function of the code is to always turn the transformer on on the positive
; half of the ac cycle and turn it off on the negative half of the ac cycle.
; Random turn on solid state relays or zero crossing relays can be used.


 title "Transformer SSR Controller"
 list p=pic12C508, st=off, x=on, n=75, r=dec
 include "p12c508.inc"
 __CONFIG B'000000001010'

;------------------------------ RAM REGISTERS -----------------------------

 cblock H'0007'
 COUNT1  ;test counter
 COUNT2  ;test counter
 endc

;---------------------------- PORT PIN ASSIGNMENTS --------------------------

#define  SSR1 GPIO,0  ;pin 7, output, SS relay 1
#define  SSR2 GPIO,1  ;pin 6, output, SS relay 2
#define  AC GPIO,2  ;pin 5, input, AC polarity, HCPL-3760
#define  TE GPIO,3  ;pin 4, input, transformer enable
#define  BLEEDER GPIO,4  ;pin 3, output, bleeder resistor
#define  ZR GPIO,5  ;pin 2, input, zero crossing or random
     ;1=Zero crossing, 0=Random turn on

;-------------------------------------------------------------------------------
 org 0
 movlw 0
 movwf OSCCAL  ;int RC oscillator calibration value

 movlw B'10000110' ;wake-up off, pull-ups on, T0 int clk,
 option   ;T0 prescaler, =128

;------------------------------- MAIN ------------------------------------
 bcf SSR1  ;set pin 7 latch low
 bcf SSR2  ;set pin 6 latch low
 bcf BLEEDER  ;set pin 3 latch low

main movlw B'00101100' ;GPIO, 0 = output pin
 tris GPIO  ;set I/O pin functions

 bcf SSR2  ;ssr2 is off
 bsf BLEEDER  ;bleeder is on
 btfss TE  ;is TE enable signal high
 goto TX0  ;no, so keep checking

 movlw 50  ;yes, make sure it was not a glitch
 movwf COUNT2
 call wait  ;wait 50ms
 btfss TE  ;is TE enable still high
 goto TX0  ;no, it was a glitch, keep waiting

 bcf BLEEDER  ;yes, turn off bleeder resistor
 clrf COUNT2  ; 1/4 second delay
 call wait  ;

 movlw 4  ;load COUNT2 with 4ms wait after zero crossing
 movwf COUNT2  ;which will be near AC peak
 call ACtrig  ;wait for +zero crossing of AC voltage
 call wait

 bsf SSR1  ;turn on SS relay 1 (soft start resistor)
 clrf COUNT2  ;load ms counter for 1/4 second delay
 call wait  ;allow power supply caps to charge

 movlw 4  ;load COUNT2 with 4ms wait after zero crossing
 movwf COUNT2  ;which will be near AC peak
 call ACtrig  ;wait for +zero crossing of AC voltage
 call wait
 bsf SSR2  ;turn on SS relay 2 (main)

;------------------------------ TURN OFF ---------------------------------
; turn off transformer at negative peak

TX1 bsf SSR1  ;ssr1 is on
 bsf SSR2  ;ssr2 is on
 bcf BLEEDER  ;bleeder is off
 btfsc TE  ;is TE enable signal low
 goto TX1  ;no, it is still high, so keep checking

 movlw 50  ;yes, make sure it was not a glitch
 movwf COUNT2
 call wait  ;wait 50ms
 btfsc TE  ;is TE enable still low
 goto TX1  ;no, it was a glitch, keep waiting

 movlw 12  ;yes, load 12ms wait after zero crossing
 btfsc ZR  ;using zero crossing SS relay?  ZR=1?
 movlw 4  ;yes, load 4ms wait after zero crossing
 movwf COUNT2  ;which will be near AC peak
 call ACtrig  ;wait for +zero crossing of AC voltage
 call wait
 bcf SSR2  ;turn off solid state relay 2
 clrf COUNT2  ;1/4 second delay
 call wait  ;
 movlw 12  ;load 12ms wait after zero crossing
 btfsc ZR  ;using zero crossing SS relay?  ZR=1?
 movlw 4  ;yes, load 4ms wait after zero crossing

 movwf COUNT2  ;which will be near AC peak
 call ACtrig  ;wait for +zero crossing of AC voltage
 call wait
 bcf SSR1  ;turn off SS relay 1

 clrf COUNT2  ;1/4 second delay
 call wait  ;

 bsf BLEEDER  ;turn on bleeder resistor
 clrf COUNT2  ;1/4 second delay for power supply bleed down
 call wait  ;
 goto main  ;wait for turn-on

;------------------------------ AC TRIGGER -------------------------------

;wait for a low  to high transition from the HCPL-3760 opto-coupler

ACtrig:
ac1 btfsc AC  ;is AC input signal low
 goto  ac1  ;no, it is high, keep waiting until low
 nop   ;yes, it is low now

ac0 btfss AC  ;is AC input signal high
 goto ac0  ;no, it is low, keep waiting until high
 nop   ;yes, it is high now
 return

;------------------------------ MS WAIT DELAY ----------------------------
;enter with milli-second value in COUNT2 register
;exits with COUNT1 and COUNT2 =0

wait:
wait1 clrf  COUNT1
wait2 nop
 decfsz COUNT1,1
 goto wait2
 decfsz COUNT2,1
 goto  wait1
 return
;-------------------------------------------------------------------------------

end
```

FIGURE 2



Using only a zero-crossing relay results in core saturation and a disastrous 46A peak current in the transformer's primary winding.

FIGURE 3



The smart-switch circuit in Figure 1 greatly reduces core saturation, resulting in well-behaved primary current. Trace R1 results from peak and same-polarity switching; trace 1 represents peak and opposite-polarity switching.

One internal diode in the optocoupler rectifies the ac signal to indicate the positive half cycle. The µC has two solid-state-relay outputs: SSR1 and SSR2. When the Transformer Enable input goes high, the µC waits 250 msec, detects the next positive edge from the optocoupler, waits 12 msec, and then turns off SSR1. SSR2 has a 250-msec delay from SSR1 and operates as a last-on, first-off output to shunt a soft-start resistor. Pin 3 is an optional output for a power-supply bleeder switch or a status indicator. (DI #2170) EDN

To Vote For This Design, Circle No. 371

EDITED BY BILL TRAVIS & ANNE WATSON SWAGER

Circuit translates A law to μ law

ROLANDO HERRERO, INSTITUTO TECNOLÓGICO DE BUENOS AIRES, ARGENTINA

Two common methods exist to compand voice for transmission through a PCM channel. In Europe, A law involves converting a 12-bit input signal to an 8-bit encoded output. In the US, μ law involves encoding 13 bits to 8 bits. You can use a translator to convert from A law to μ law (**Figure 1**). The converter is asynchronous and requires only an 8-bit A law input to provide an 8-bit μ law output.

In A law, the input level divides into eight regions in which a uniform 4-bit conversion takes place. Regardless of the region, the output encodes 16 possible values. Each region corresponds to a segment in **Figure 2**, and the lower values have a better resolution (this **figure** shows only segments 0 through 5). To encode the input takes 8 bits; 4 bits indicate the uniform converted value in the segment, and the other 4 bits divide to represent the segment value itself (S0 to S7, coded with 3 bits) and whether the signal is positive or negative (1 bit).

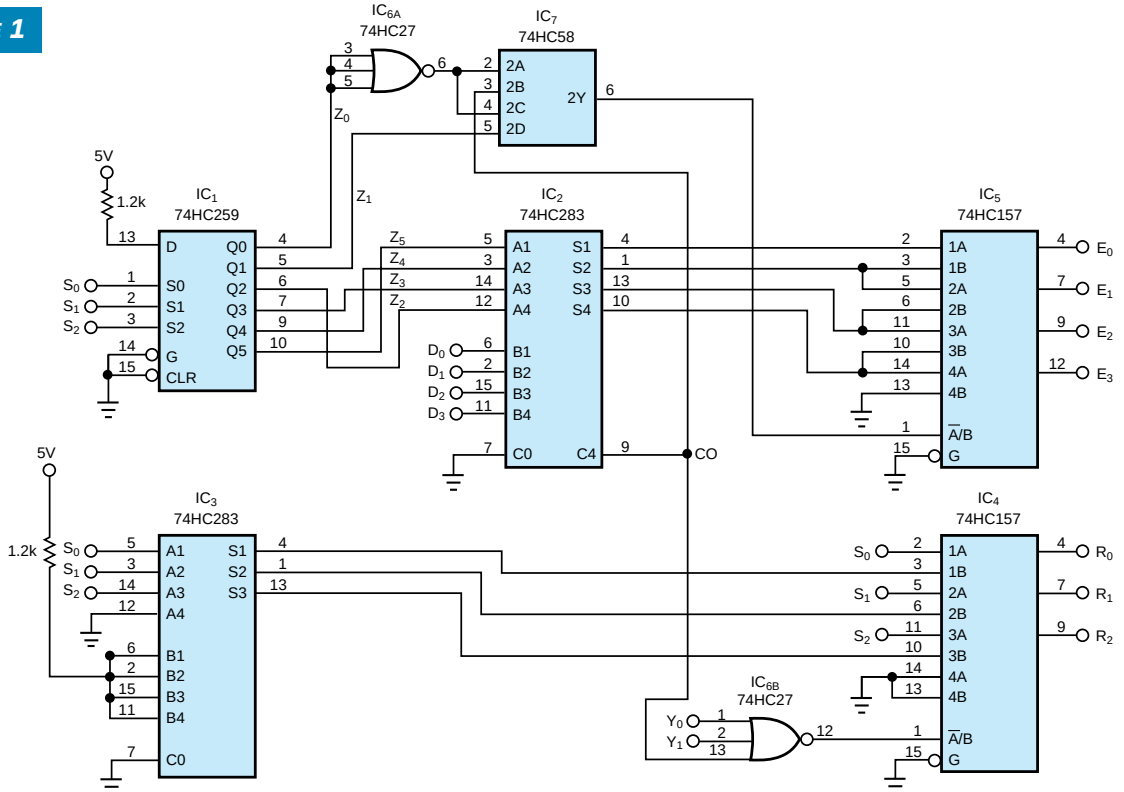
Alternatively, with μ law, also included in **Figure 2**, all but the first segments have a wider dynamic range and thus more spaced quantization levels (for 4 bits) compared with A law. Instead of 12 bits, 13 bits imply a wider dynamic range but a worse resolution for low input levels.

Figure 2 also illustrates the loss of resolution when converting the output A of A law to output A' of μ law. Depending on the law, either 8 bits (A law) or 4 bits (μ law and higher quantization levels) represent the value, therefore, the transitions occur faster around output A than around output A'. For the A to A' translation, the slope of A law is twice the slope of μ law.

Although information loss occurs during the conversion of value A, the same is not true for B. For B, the A law and μ law slope are the same, and the quantization level is the same. Thus, the difference between B and B' involves only a translation and a change of segment (B in S4, B' in S3). A simple comparison shows that the A value suffers a translation and a loss of information but remains in the same segment after conversion.

The design of the encoder must take into account the A law signal's segment and offset value, as does the following algorithm for which the A law input signal is PSD, and the μ law output signal is QRE, for which P,Q=polarity (1 bit), S,R=segment (3 bits) and D,E=value (4 bits):
If S=0, then Q=P, R=S, and E=D.
If S=1, then Q=P, R=S, and E=D/2.

FIGURE 1

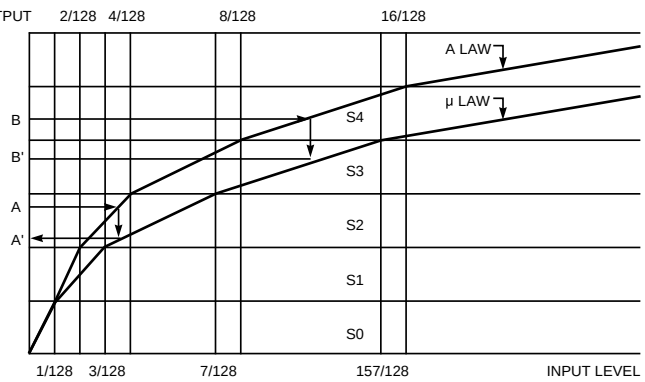


This A law-to-μ law translator inputs values of S and D and outputs E and R according to a specific algorithm.

If S=2 and D<8, then Q=P, R=S–1, and E=D+8.
If S=2 and D>7, then Q=P, R=S, and E=(D–8)/2.
If S=3 and D<12, then Q=P, R=S–1, and E=D+4.
If S=3 and D>11, then Q=P, R=S, and E=(D–12)/2.
If S=4 and D<14, then Q=P, R=S–1, and E=D+2.
If S=4 and D>13, then Q=P, R=S, and E=(D–14)/2.
If S=5 and D<15, then Q=P, R=S–1, and E=D+1.
If S=5 and D>14, then Q=P, R=S, and E=(D–15)/2.
If S=6, then Q=P, R=S–1, and E=D.
If S=7, then Q=P, R=S–1, and E=D.

According to this algorithm, the conversion requires both addition and subtraction, depending on S and D. You can express each subtraction as an addition to implement both in the same circuit. Thus, you can express the algorithm as follows, where CO=Carry out:
If S=2 and D<8, then Q=P, R=S–1, Z=8, and E=D+Z (CO=0).
If S=2 and D>7, then Q=P, R=S, Z=8, and E=(D–8)/2=(D–16+Z)/2=(D+Z)/2 (CO=1).
If S=3 and D<12, then Q=P, R=S–1, Z=4, and E=D+Z (CO=0).
If S=3 and D>11, then Q=P, R=S, Z=4, and E=(D–12=D–16+Z)=(D+Z)/2 (CO=1).
If S=4 and D<14, then Q=P, R=S–1, Z=2, and E=D+Z (CO=0).
If S=4 and D>13, then Q=P, R=S, Z=2, and E=(D–14)/2=(D–16+Z)/2=(D+Z)/2 (CO=1).
If S=5 and D<15, then Q=P, R=S–1, Z=1, and E=D+Z (CO=0).
If S=5 and D>14, then Q=P, R=S, Z=1, and E=(D–15)/2=(D–16+Z)/2=(D+Z)/2 (CO=1).
The value of Z depends on S: $Z=2^{5-S}$. Once you define Z, the algorithm performs the same D+Z operation for each S. The carry-out (CO) signal determines whether R is equal to S or S–1. Therefore, this implementation simultaneously solves two problems. Furthermore, the same technique applies for S=6 and S=7, when Z=0.

FIGURE 2



Converting output A of A law to A' of μ law incurs a loss of information. However, no information loss occurs when converting from B to B', because the slopes of the two curves are the same at that point.

In **Figure 1**, a 3×8 decoder, IC_1, converts S to Z, which IC_2 adds to D. If the CO is a 1, E is (D+Z)/2; otherwise, R is S–1. To choose between both options, the circuit uses the CO signal to control data selectors IC_4 and IC_5. These devices select between two possible outputs: S or S–1 and D+Z or (D+Z)/2, respectively. A second adder, IC_3, implements S–1 by summing the S inputs with 15. The circuit derives (D+Z)/2 by shifting D+Z into the inputs of data selector IC_5. Additional logic ensures that no conversion occurs when S=0 and that E=D/2 when S=1.

The 8-bit input is P0/S2/S1/S0/D3/D2/D1/D0, and the 8-bit output is P0/R2/R1/R0/E3/E2/E1/E0. The schematic doesn't show P0 because this parameter's value doesn't change. The circuit was tested with a Motorola (www.mot.com) MC145554 μ law PCM codec-filter and an 8TR641 (AT&T, www.att.com) E1 multiplexer. (DI #2192)

To Vote For This Design, Circle No. 412

MCS-51 endows MicroLan-like protocol to UARTs

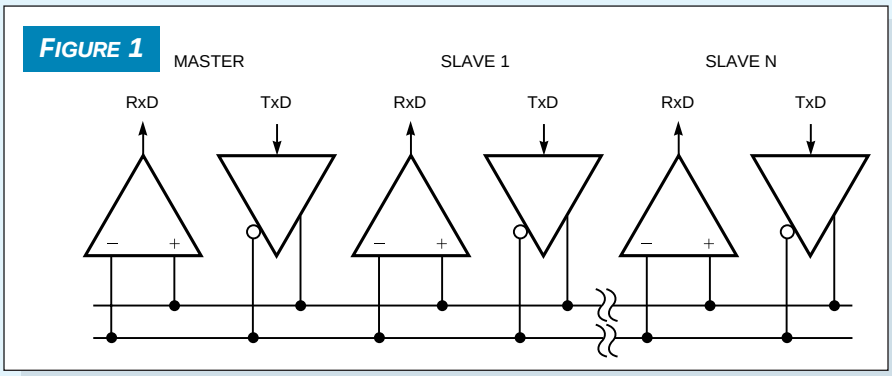
SK SHENOY, NPOL, KOCHI, INDIA

μCs such as the 8051 and 8096 and UARTs such as the 82510 provide hardware support for a multiprocessor asynchronous serial-communication protocol (MicroLan). This feature is useful in applications in which a number of processors interconnected in a multipoint configuration jointly perform a task, with a master processor controlling slaves by sending data or commands in a selective manner (**Figure 1**). The protocol operates as follows:

When the master wishes to transmit a block of data to a slave, it first sends an address byte that identifies the slave. All data and address bytes are nine bits long. An address byte differs from a data byte in that its ninth bit is one (for a data byte it's zero). The communication subsystem normally initializes in a mode where the serial-port interrupt activates only when the ninth bit is one. Thus, no slave receives an interrupt from a data byte. An address byte, however, interrupts all slaves, which then examine the received byte. Next, the addressed slave switches to a mode in which data bytes

also receive interrupts, while other slaves go about their business uninterrupted by the data transfer. The address bytes thus control the data flow into a particular node. Indication of the end of a data block can come from either sending a data-length field at the beginning of the block or from the receipt of another slave or reserved address.

The crucial requirement for realizing the protocol is a means of distinguishing address from data bytes. You can effect this identification in many popular UARTs by using an obscure feature found in most UARTs: the capability to transmit and recognize (with an interrupt on) the break condition. This condition is nothing but a "space," or low, in the transmit line, of a duration equal to or greater than an entire asynchronous character-transmission time, including stop bits (**Figure 2**). In this scheme, the whole data block (including address) from a master is sandwiched between break characters to form a data "frame" (**Figure 3**), and the address byte is recognizable as the one that immediately follows a break character.

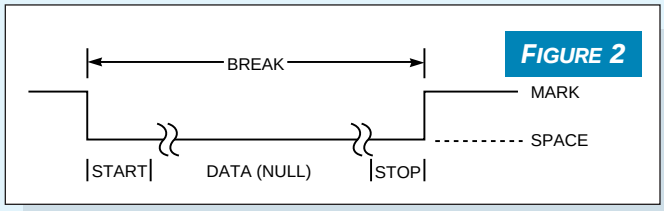


A master-slave arrangement uses RS-422 transceivers to effect a multipoint data-transfer configuration.

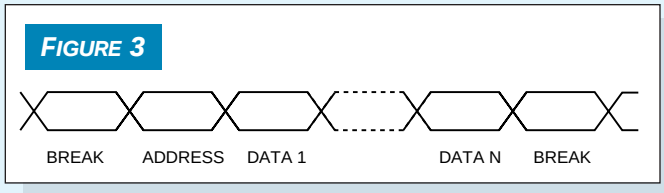
The Turbo C program in **Listing 1** demonstrates the transfer of variable-size messages between two PCs (with 8250-compatible UARTs) using the method described here. **Figure 4** shows the 8250 register formats. The procedure works with most other UARTs. You can download the file from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the "Software Center" to download the listing from DISIG #2193. A null-modem cable interconnects the PCs' COM ports. The destination PC accepts only the messages addressed to it. Note that, although the PCs here interconnect in a point-to-point manner, usually the stations interconnect using balanced RS-422 or tristate drivers in a multipoint configuration, as in **Figure 1**.

A global variable, Receive_Count, initialized to zero, handles frame reception. Initially, the protocol enables only receive-error interrupts. Each time the routine detects a break, the UART raises a receive-error interrupt, and the ISR (interrupt service routine) then enables the receive-data interrupts. On subsequent receive interrupts, if Receive_Count is zero, the ISR checks if the first address byte matches the station's address. If not, the receiver goes back to the initial waiting state, with the receive-error interrupts enabled and the receive-data interrupts disabled, such that the routine ignores the subsequent data bytes. If an address match occurs, the ISR stores the subsequent incoming data bytes in the receive buffer, with Receive_Count as index. If Receive_Count is nonzero when the break interrupt occurs, it is an end-of-frame break. Then the routine calls the frame-processing function, Receive_Count resets to zero, and the receiver again reverts to the initial waiting state.

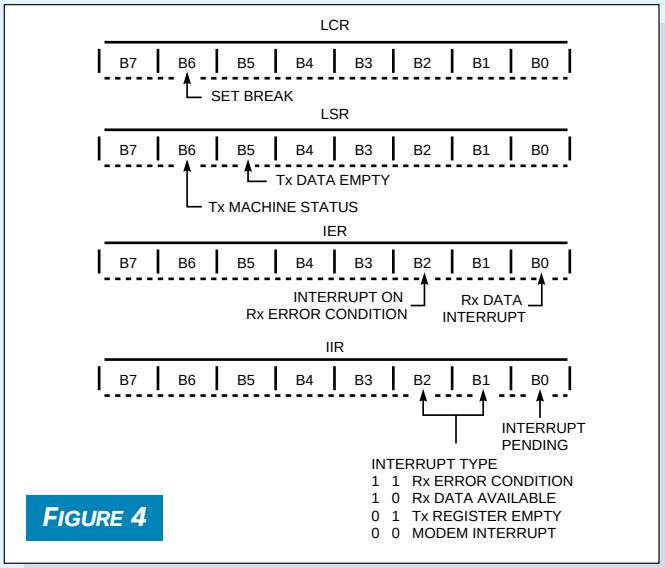
To transmit a break, the protocol sets bit 6 (set break) of the line-control register to one. The UART then takes its trans-



The ability to recognize the break condition is key to the master-slave transfer protocol.



This n-byte data frame shows the data block sandwiched between break characters.



These 8250 register formats demonstrate the multipoint-transfer protocol.

mission line low until bit 6 receives a zero. To make the duration of the break equal to one character-transmission delay, the routine transmits a null (00 hex) character. Bit 6 of the line-control register (transmit machine status) indicates when this delay is over; the break bit then resets. To enable detection of the break, bit 2 of the interrupt enable register (interrupt on receive error condition) sets during UART initialization. Bit 0, set to one, enables receive data interrupts. In the ISR, bits 1 and 2 of the interrupt-identification register indicate the interrupt type.

In this scheme, no CPU overhead is wasted examining each character to detect addresses/packet boundaries. Also, a slave must process only three interrupts per data packet transmitted on the bus, and blocks of data not addressed to the slave do not disturb it. Because the break is not a legitimate data character, it is data transparent; you can use it for binary-data exchange. The packet-boundary detection is immune to data errors. You can make it even more robust by including data-length and check-sum fields in the frame to enable error detection. You can also use parity error detection. Note that the method can support broadcast/multicast message transfer by designating some addresses for these purposes. You can also implement any-node-to-any-node communication by polling the master, as in the SDLC protocol. (DI #2193)

To Vote For This Design, Circle No. 413

LISTING 1–TRANSFER OF VARIABLE-SIZE MESSAGES BETWEEN TWO PCS

```
    #include <stdio.h>
    #include <stdlib.h>
    #include <conio.h>
    #include <dos.h>


    /* COM PORT DEFINITIONS AND GLOBAL VARIABLES */
    #define      com_reg     0x3f8  /* Default is com1; 2f8 for com2 */
    #define      DATA_PORT   com_reg + 0
    #define      LINE_CNTRL  com_reg + 3
    #define      MODEM_CNTRL com_reg + 4
    #define      INT_ENABL   com_reg + 1
    #define      INT_IDENT   com_reg + 2
    #define      LINE_STS    com_reg + 5
    #define      MODEM_STS   com_reg + 6
    #define      BAUD_LOW    com_reg + 0
    #define      BAUD_HIGH   com_reg + 1
    #define      DLAB_SET    0x80
    #define      BAUDMSB     0
    #define      BAUDLSB     0xc /* 9600 BPS */
    #define      CNTRL_CMD   7   /* 8 BIT, 2 STOP BIT, NO PARITY */
    #define      WAIT_TX_RDY()  while (((inportb(LINE_STS))&0x60)!=0x60)
/* Check for Tx buf empty & Tx shift reg empty */


    unsigned char sdatabuf[256],rdatabuf[256]; /* Send & Recv buffers */
    int  Receive_Count = 0;  /* Counter for data stored in rdatabuf[] */
    void interrupt(*OldComHandler)(void);
    unsigned char Myaddr,Txaddr;



    void processdata(void) /* TO DISPLAY RECEIVED DATA PACKET */
    {
      int i;
      cprintf("\n\rRX Data > "); clreol(); /* Received data cursor */
      for (i = 1; i< Receive_Count; i++) /* Leaving out Addr byte */
    putch(rdatabuf[i]); /* Display received data */
      cprintf("\n\r"); /* New line */
      clreol(); /* Clear line */
    }

    void interrupt service_sio(void) /* ISR: TAKES CARE OF PACKET RECEPTION */
    {
      unsigned char  iir ;

      iir = (inportb(INT_IDENT) >> 1) & 3; /* Get interrupt type */
      switch(iir)
      {
    case 0: /* Modem status int DSR,CTS,RI,RLSD */
      inportb(MODEM_STS); /* Ignore; reading IIR resets int */
      break;/* reading IIR resets int */
    case 1: /* Tx int */
      break;/* reading IIR resets int */
   case 2: /* Rx int */
     rdatabuf[Receive_Count++] = inportb(DATA_PORT); /* Store packet data */
     if((Receive_Count == 1) && (rdatabuf[0] != Myaddr))
     /* If First Address) byte but no address match */
     {
outportb(INT_ENABL,0x4); /* IER; enable Only Rx Machine error int */
Receive_Count = 0;
     }
     break;
   case 3: /* Rx error (Break detect etc.) */
     inportb(DATA_PORT); /* Read Null char */
     if(((inportb(LINE_STS))&0x10) == 0x10)
     /* Break detected; Reading LSR Resets int */
     {
 if(Receive_Count) /* Complete Frame Over */
 {
    processdata(); /* Process the frame  */
    outportb(INT_ENABL,0x4); /* IER; enable only Rx Machine error int */
 }
 else outportb(INT_ENABL,0x5); /* IER; enable RX Data int also */
 Receive_Count = 0; /* Reinitialize for next frame */
     }
     }
     outportb(0x20,0x20); /* EOI to 8259 PIC */
     return;
   }

   void init_serial_io(void) /* TO INITIALISE SERIAL PORT */
   {
     outp(LINE_CNTRL,DLAB_SET);   /* DLAB_SET */
     outp(BAUD_LOW,BAUDLSB); outp(BAUD_HIGH,BAUDMSB); /* 9600 BAUD */
     outp(LINE_CNTRL,CNTRL_CMD);  /* 8 BIT,2 STOP BIT,NO PARITY */
     outp(MODEM_CNTRL,8);         /* DTR,RTS & OUT2 SET */
     OldComHandler = getvect(0xc);/* 0xb for com2 */
     disable();
     setvect(0xc,(service_sio)); /* 0xb for com2 */
     outportb(0x21,((inportb(0x21))&(!0x10)));/* PIC mask word 0x8 for com2 */
     outportb(INT_ENABL,0x4); /* IER; enable only Rx Machine error int */
     enable();
 }

 void SendBreak(void) /* TO TRANSMIT A BREAK OF ONE CHARACTER DURATION */
 {
     outportb(LINE_CNTRL,inportb(LINE_CNTRL) | 0x40); /* LCR; set break */
     outportb (DATA_PORT,0); /* Send NULL data */
     WAIT_TX_RDY(); /* Wait on TxShift Reg Empty; Null char is shifted out  */
     outportb(LINE_CNTRL,inportb(LINE_CNTRL) & 0xbf); /* LCR; remove break */
 }


 /* TO TRANSMIT A DATA PACKET */
 void SendBuffer(unsigned char packet[],int DatLen)
 {
    int i;
    SendBreak(); /* Send START OF PACKET break */
    WAIT_TX_RDY(); /* Wait for Tx Ready */
    outportb (DATA_PORT,Txaddr); /* Send Tx address */
    for (i=0; i<DatLen; i++) /* For each message byte */


      {
WAIT_TX_RDY(); /* Wait for Tx Ready */
outportb (DATA_PORT,packet[i]); /* Send next data char */
      }
      WAIT_TX_RDY(); /* Wait for Tx Ready */
      SendBreak(); /* Send END OF PACKET break */
      WAIT_TX_RDY(); /* Wait for Tx Ready */
  }



  unsigned char getaddr(char* mess) /* TO READ AN ADRRESS FROM THE CONSOLE */
  {
     unsigned char c, databuf[100];
     int addr,count = 0;

     cprintf("\n\r%s",mess); /* Prompt for input */
     clreol();
     while(1) /* Forever Loop */
     {
 if((c=getche()) == 27) exit(0); /* Exit if Escape key pressed */
 databuf[count++] = c; /* Get typed characters into the buf */
 if(c == '\r')  /* If Enter Key pressed */
 {
    if((sscanf(databuf,"%d",&addr) != 1)||((addr > 0xff)||(addr < 0)))
    { /* Read and check the string in the buffer for validity */
putch(7); /* Bell */
cprintf("\r\nError: Type in a number between 0 and 255");
cprintf("\n\r%s",mess); /* Transmit Prompt */
clreol(); /* Clear to end of line */
count = 0;
     }
     else break;
 }
     }
     return((unsigned char)addr);
  }


  void restoreint(void) /* FUNCTION WHICH DOES THE CLEAN-UP AT EXIT TIME */
  {
     setvect(0xc,(OldComHandler)); /* Restore int vector; 0xb for com2 */
     outportb(0x21,((inportb(0x21))|(0x10)));/* PIC mask word 0x8 for com2 */
  }
```

Voltage monitor prevents deep discharge of battery

ROGER KENYON, MAXIM INTEGRATED PRODUCTS, SUNNYVALE, CA

The circuit in **Figure 1** monitors battery voltages from 2.7 to 5.5V while drawing less than 25 µA. When the voltage reaches a minimum threshold established by R_1 and R_2, (V_{TH}=2.63V for the values shown), the high-side switch (IC_2) turns off and disconnects the battery from the load.

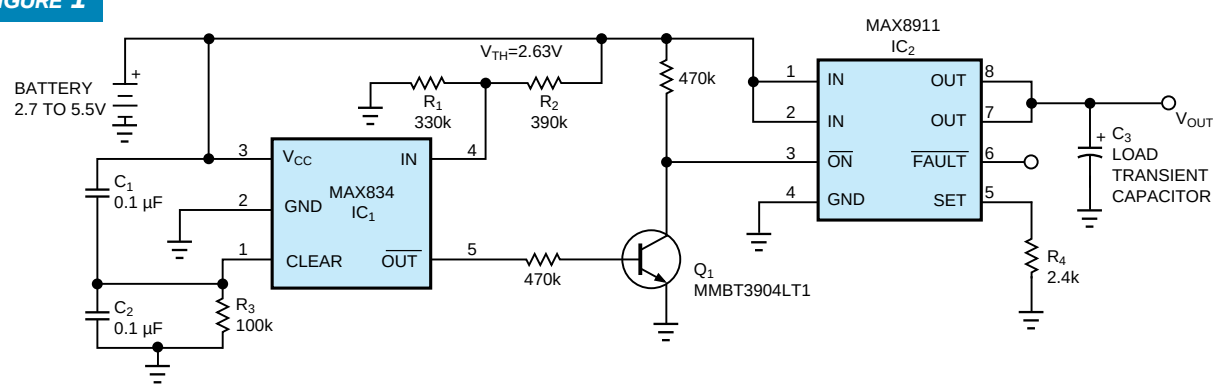
IC_1 is a voltage monitor with an open-drain latched output. Normally high, the output latches low when the battery voltage drops below V_{TH}. Once triggered, the output remains low even when the now-unloaded battery voltage rebounds to a level above V_{TH}. This behavior prevents the oscillation that would otherwise occur as connect/disconnect action causes the battery voltage to fluctuate. To reset the latch, the CLEAR input must go high for a minimum of 1 µsec.

The $C_1/C_2/R_3$ network applies the latch-clearing pulse when you connect a new battery. Rechargeable-battery applications require other schemes for clearing the IC_1 output, such as an spst momentary pushbutton switch (**Figure 2a**) or simply a connection via the battery-charger connector (**Figure 2b**).

To set a different value of V_{TH}, choose a convenient value for R_1, and then calculate R_2: $R_2=R_1\times V_{TH}/(1.204–1)$. IC_2 limits its switch current at a level that the value of R_4 determines: $I_{LIMIT}=1240/R_4$, where R_4 is in ohms and I_{LIMIT} is in amperes, with a maximum of 1A. For the R_4 value in **Figure 1**, this limit is 500 mA. (DI #2191)

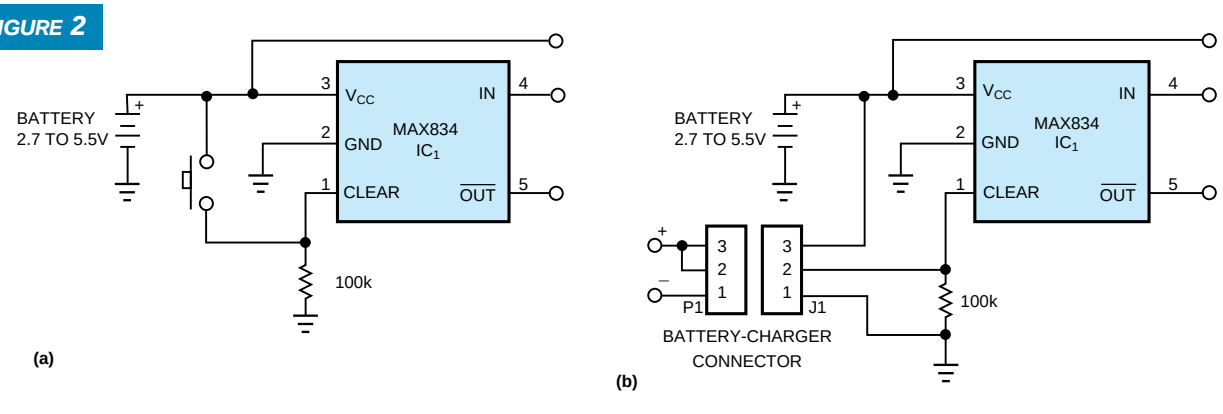
To Vote For This Design, Circle No. 414

FIGURE 1



When the voltage reaches a minimum threshold established by R_1 and R_2, the high-side switch, IC_2, turns off and disconnects the battery from the load.

FIGURE 2



Other schemes for clearing IC_1's output include an spst momentary pushbutton switch (a) and a connection through the battery-charger connector (b).

Add switch-and-LED I/O to DSP's serial port

STAN SASAKI, TWENTY-FIRST DESIGNS, LAKE OSWEGO, OR

When you debug an embedded DSP design, it's handy to have a bank of switches and LEDs to simulate inputs or to display intermediate results. You can attach 16 switches and 16

A bank of switches and LEDs simulates inputs and displays intermediate results as an aid to debugging embedded-DSP designs.

FIGURE 1

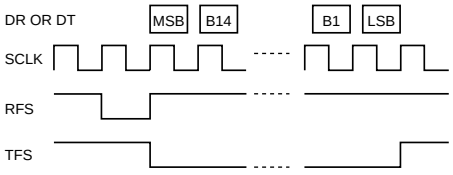
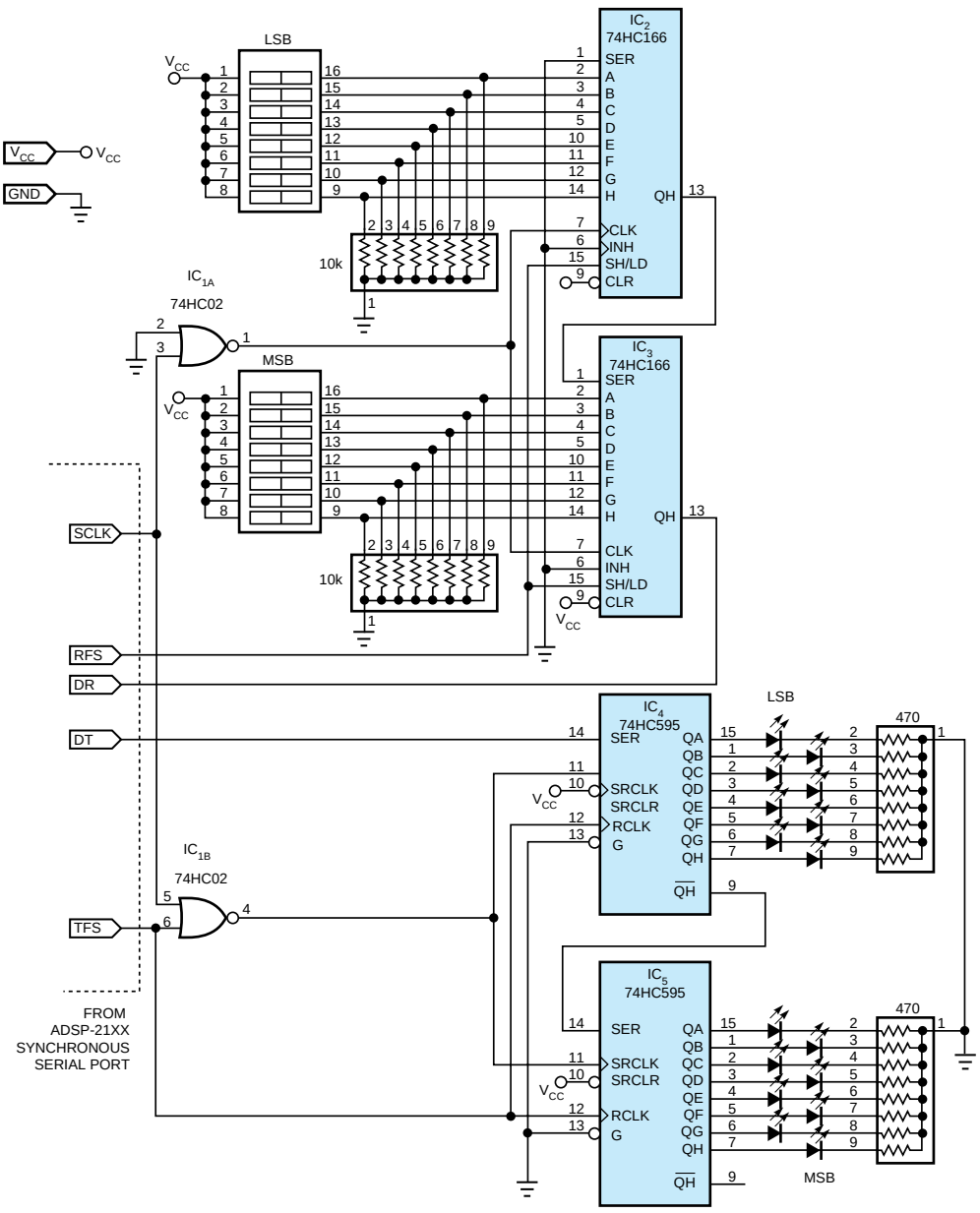


FIGURE 2



The DSP shifts data, MSB-first, in each direction. The data is valid on the falling edge of SCLK.

LEDs to the data bus using octal latches and decoded address strobes. However, this connection can be a wiring nightmare—or even impossible if the design does not generate the required control strobes. Most DSP chips have a synchronous serial port that provides high-speed (>10 Mbps) bidirectional communication over five wires. The circuit in **Figure 1** provides 16-bit switch and LED I/O using the on-chip serial port of Analog Devices' ADSP21xx family. Some ADSP devices, such as the ADSP2105, have only a single serial port that the target application can use. However, during debug, you can use the pin-compatible ADSP2115, which has two serial ports. You can bring the five signals from the second serial port to a header that attaches to the circuit in **Figure 1**.

The DSP generates a continuous serial clock (SCLK) in **Figure 2**. Data shifts, MSB-first, in each direction and is valid on the falling edge of SCLK. For transmission to the LEDs, the DSP asserts the transmit-frame-sync (TFS) line while the 16 bits clock out on DT (data transmit). Cascaded 8-bit serial-to-parallel shift registers, IC_4 and IC_5, capture the data, and the LEDs receive an update on the rising edge of TFS. For switch reception, the DSP generates a receive-frame-sync (RFS) pulse one bit-time before reading the data. RFS latches the switch states into cascaded parallel-to-serial shift registers, IC_2 and IC_3, and the data shifts into the serial port over DR (data receive) on the next 16 SCLK cycles. You can configure the serial port to generate the RFS pulse at a rate divided down from the SCLK frequency.

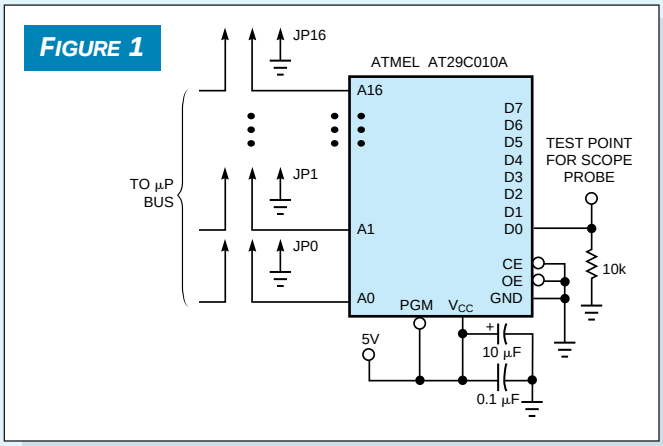
A single instruction reads the switches or writes to the LEDs. For example, the instruction AR=RXn transfers the last switch value from the receiver of serial port n into the 16-bit accumulator, AR. Note that n can be either 0 or 1, depending on which on-chip serial port you use. Similarly, the instruction TXn=AR transfers the 16-bit value in AR to the transmitter of serial port n, and on to the LEDs. You must configure the serial port to operate with the frame-sync types and signal polarities the circuit expects. In this case, the 16-bit value you must write to the serial port n control register is 6FCF. Because 16 LEDs can draw more than 100 mA, you may need to provide external power if the circuit under test cannot supply the necessary current. Because HC parts operate at 3 and 5V, this circuit works for both 3 and 5V ADSP devices. However, for 3V operation you may need to lower the SCLK speed to meet HC performance. (DI #2196)

To Vote For This Design, Circle No. 415

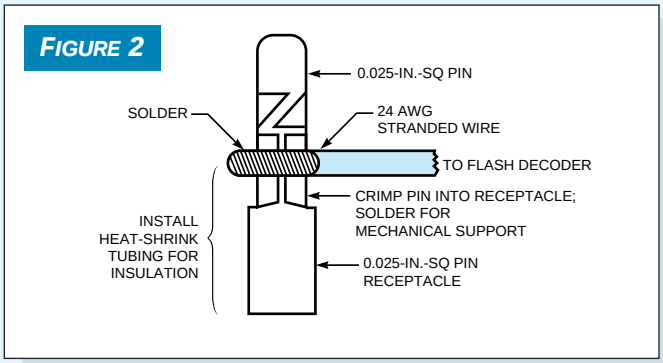
DSO-triggering scheme is cheap and efficient

ROBERT PERRIN, Z-WORLD, DAVIS, CA

Although a logic analyzer is useful in troubleshooting a new design, nothing beats a digital storage oscilloscope (DSO) for the ability to see bus levels and timing. However, the trigger mechanisms on most DSOs are not sophisticated enough to trigger on a specific µP bus state. For example, you may need the DSO to trigger on a specific address while the µP is attempting to read from it. This operation would correspond to the start of some code segment you're interested in observing. Most DSOs cannot provide such triggering. **Figure 1** shows a circuit that generates a trigger on a specific µP bus condition.



A flash chip with all addresses but one filled with zeros dramatically improves DSO-triggering capabilities.



This test pin plugs into a 0.025-in. test point while providing another 0.025-in. pin for other test equipment.

The flash IC serves as a decoder. The entire address space is filled with zeros, except for a single one at the address that corresponds to the µP bus state of interest. The DSO can trigger on the D0 line of the flash chip, and begin recording events at the bus cycle of interest. Shorting blocks, used with JP0 through JP16, configure the address pins corresponding to the trigger condition. The µP bus signals required to detect the trigger condition attach to the open address lines in the flash IC. You can program multiple bus conditions to generate a trigger on the same data line. Other data lines can generate additional trigger conditions.

The design in **Figure 1** uses a 70-nsec flash chip. The speed is actually irrelevant, because DSOs can acquire and display pretrigger information. The absolute timing of the trigger is arbitrary. The relative timing of the bus signals is the parameter of interest. The DSO displays pretrigger and posttrigger data properly. As a result, you'll have a solid picture of the bus signals for evaluation. The design in **Figure 1** uses a ZIF socket to hold the flash IC; thus, reprogramming is easy. In addition, the test pins in **Figure 2** make it easy to grab signal lines from PLCC test clips, while providing a 0.025-in. square post pin to which the logic analyzer can connect. (DI #2197)

To Vote For This Design, Circle No. 416

Piezo device generates buzz, beep, or chime

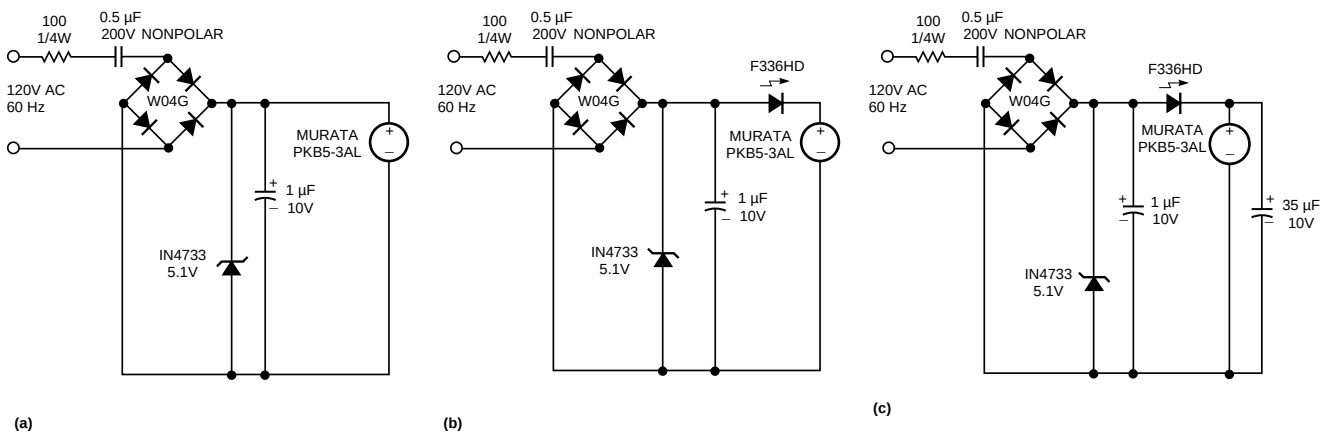
DENNIS EICHENBERG, PARMA HEIGHTS, OH

Piezoelectric buzzers, such as the Murata (Smyrna, GA) PKB5-3A in **Figure 1**, make excellent alarms. They're compact, lightweight, efficient, and reliable. However, a piezo alarm is a dc device; it requires additional circuitry to operate from an ac source. The circuits in **Figure 1** provide a simple and inexpensive way to obtain the dc drive. The W04G full-wave bridge rectifier produces a full-wave dc waveform from the 120V ac line. The 100Ω resistor protects the circuit from surges when you first apply power. The 1N4733 5.5V zener diode protects the buzzer against high-voltage excursions. The 1-µF capacitor provides filtering for the buzzer.

The circuit in **Figure 1a** produces a true buzzer sound. The addition of an F336HD flashing LED (part number 276-036 at Radio Shack) in **Figure 1b** changes the alarm to a beeper, and it also provides a visual alarm. The LED produces a constant pulse of light at approximately 1 Hz without the addition of a time-constant capacitor. The LED starts immediately when you apply power, and it's insensitive to temperature variations. The addition of a 35-µF capacitor in parallel with the buzzer (**Figure 1c**) changes the audible alarm to a pleasing chime. The value of the capacitor is not critical; you can obtain various sound effects by varying it. (DI #2194)

To Vote For This Design, Circle No. 417

FIGURE 1



A handful of inexpensive components configures a piezo alarm device as a buzzer (a), a beeper (b), or a chime (c).

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____ Fax _____

E-mail _____

My e-mail address may be published Yes____ No____

Company _____

_____

Address _____

_____

Country _____ ZIP _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

The winning Design Idea for the September 25, 1997, issue is entitled "Single chip builds tiny aircraft receiver," submitted by Steven Hagemanr of Hewlett-Packard (Santa Rosa, CA).

EDITED BY BILL TRAVIS & ANNE WATSON SWAGER

25-kV generator tests insulation

LUKASZ ŚLIWCZYŃSKI AND PRZEMYSLAW KREHLIK, UNIVERSITY OF MINING AND METALLURGY, KRAKÓW, POLAND

To generate high voltages with proper insulation between the "hot node" and the rest of the circuitry, a car ignition coil can function in place of a high-voltage transformer. These coils have voltage ratings of approximately 20 kV, so you can use them to produce voltages around this value. Because you know the turns ratio of the coil, you can make a stable high-voltage source using a well-controlled voltage at the primary side (**Figure 1**).

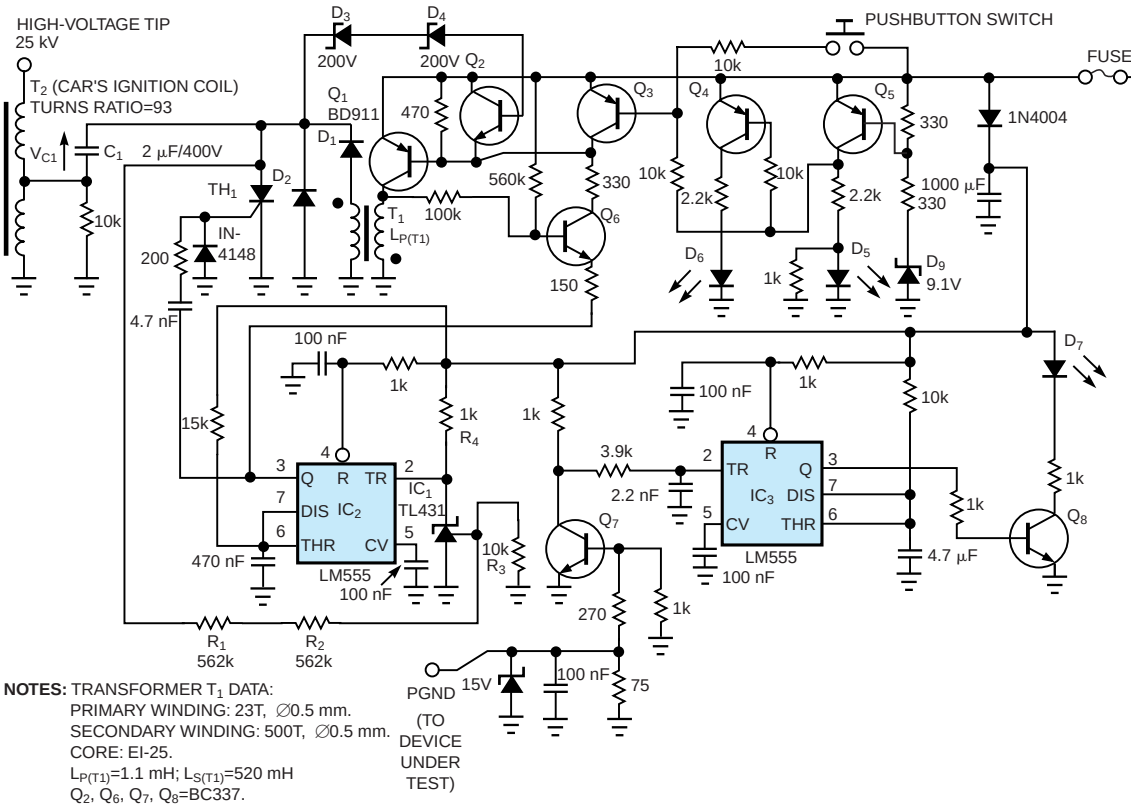
A high-voltage source is a useful device for many applications, including when it is necessary to evaluate the continuity of the dielectric coating deposited on a metal surface. If you also want to estimate the breakdown strength of the coating, this voltage source must be stable. You can easily generate the high voltage with the use of a step-up transformer, but the serious problem of proper insulation emerges. For voltages greater than a few kilovolts, specially constructed transformers with the old insulation are often useful, but these devices are rather expensive and bulky.

The main part of the generator in **Figure 1** consists of a free-running converter comprising Q_1, Q_6, and the transformer, T_1. During the first part of the conversion cycle, Q_1 is saturated, and energy stores in the magnetic field of T_1. D_1 is reverse-biased during this time. In the second part of the cycle, Q_1 is in cutoff, and the current from the secondary winding of T_1 forces D_1 into conduction. During this time, energy pumps into C_1 through part of the ignition coil, T_2. This process allows the voltage, V_{C1}, on C_1 to build gradually in a quantized manner. The value of the individual "quantum," ΔV_{C1}, is not constant and depends on the initial voltage, V_{CO}, which comes from the previous cycle, as follows:

where

is the energy stored in the magnetic field of T_1 in the first cycle

FIGURE 1



NOTES: TRANSFORMER T_1 DATA:
PRIMARY WINDING: 23T, ∅0.5 mm.
SECONDARY WINDING: 500T, ∅0.5 mm.
CORE: EI-25.
$L_{P(T1)}$=1.1 mH; $L_{S(T1)}$=520 mH
Q_2, Q_6, Q_7, Q_8=BC337.
Q_3, Q_4, Q_5=BC307.
D_1, D_2=FR307 (FAST RECOVERY TYPE).
TH_1=MCR106-8, 2N6241 (600V, 4A).

Using a car's ignition coil produces a test voltage as high as 25 kV for insulation testing.

and $I_{CMAX(Q1)}$ is the collector current of Q_1 at the end of the first cycle. For the component values in **Figure 1**, $\Delta E_C \approx 0.5$

$$\Delta V_{C1} = V_{C0}\left(\sqrt{1+\frac{2\Delta E_C}{V_{C0}^2 \bullet C_1}}-1\right) \approx \frac{\Delta E_C}{V_{C0} \bullet C_1},$$

$$\Delta E_C = \frac{L_{P(T1)} \bullet I_{CMAX(Q1)}^2}{2}$$

mJ, and $I_{CMAX(Q1)} \approx 1A$.

R_1, R_2, and R_3 divide down V_{C1}. When this reduced voltage reaches 2.5V, the TL431's 2.5V reference starts to sink the current through R_4, so the voltage at the trigger input of the one-shot, IC_2, rapidly decreases. An output pulse from IC_2 stops the converter for about 8 msec; the emitter node of Q_6 goes high, driving it into cutoff. The rising edge of IC_2's output pulse also triggers thyristor TH_1. The thyristor connects C_1, which is charged to the appropriate voltage, directly to the primary winding of the ignition coil, and the high-voltage pulse appears at the "hot" end of the coil. A damped oscillation also starts because the ignition coil and C_1 form a resonant circuit.

When a path between the "hot" end and ground exists, part of the energy from the capacitor disperses in the electric arc, and the rest returns to the capacitor through D_2. When there is no path from this end of the current to flow, almost all of the energy pumps back into C_1. This scheme provides the circuit with relatively high efficiency.

You can calculate the voltage at the "hot" side using the following formula:

where $N_{SEC(T2)}/N_{PRI(T2)}$ is the turns ratio of the ignition coil, which equals 93 in this case. Changing the value of R_3 conveniently regulates V_{HIGH}. The accuracy of this voltage is in

$$V_{HIGH} = 2.5\left[1+\frac{R_1+R_2}{R_3}\right]\left[1+\frac{N_{SEC(T2)}}{N_{PRI(T2)}}\right],$$

the range of one "quantum" ΔV_{C1} multiplied by T_2's turns ratio. Thus, ΔV_{C1} should be small to achieve good stabilization. On the other hand, a smaller value increases the time between subsequent high-voltage pulses. In this case, the accuracy estimate of the high-voltage pulse is better than 0.5% at 25 kV.

The free-running frequency of the converter depends on the time it takes to lead Q_1 out of saturation (first part of the cycle) and the time when the current from the secondary winding of T_1 drops to a value near zero (second part of the cycle). This circuit doesn't tightly control this frequency, which isn't a critical design parameter; the values in **Figure 1** set the frequency to approximately 6 kHz.

Q_2, D_3, and D_4 prevent V_{C1} from exceeding about 400V, which protects the generator from producing excessively high voltages. Q_3, Q_4, Q_5, and associated circuitry allow for blocking the converter when the power supply to the circuit is too low. A too-low power-supply level may lead to an output-pulse amplitude from IC_2 that is too low to trigger the thyristor, so V_{C1} may reach a very high value, limited only by the breakdown voltage of the thyristor. This breakdown voltage is the second level of protection, but you can never take too much care in circuits like these.

Two LEDs indicate the status of the power supply: D_5 indicates that the level is OK, and D_6, that the power supply is too low. One-shot IC_3, Q_4, and associated components form the source of an alarm, indicated by a flashing D_7, when the isolation breaks down or a discontinuity occurs. A simple push-button switch turns on the generator.

For the component values in **Figure 1**, the circuit generates 25-kV pulses with a repetition rate of approximately 0.2 sec. This repetition rate depends on the occurrence or lack of occurrence of the electric arc. Because the amount of energy stored in C_1 is relatively low, the energy of the high-voltage pulse is also low, which is good for safety purposes. Note that it is very important and absolutely necessary to connect the part you're testing to the PGND point, because the risk of electric shock exists. (DI #2199) e

To Vote For This Design, Circle No. 414

Scheme speeds access to μP's real-time clock

JERZY CHRZASZCZ, WARSAW UNIVERSITY OF TECHNOLOGY, WARSAW, POLAND

The DS5000T (Dallas Semiconductors, www.dalsemi.com) is an 8051-compatible processor that integrates nonvolatile memory and a real-time clock. This module has an impressive set of functional extensions and security features, which makes it particularly useful for all-in-one embedded systems.

Unfortunately, access to the real-time clock is complicated and thus inefficient. You access the on-chip real-time-clock registers serially in secondary address space by selecting the ECE2 bit in the MCON register. Instead of just moving the data, you must execute MOVX instructions with appropriate address patterns. First, a 64-bit key is necessary to open the clock, followed by a read or write of the next 64 bits of date/time data. The access routines available from the manufacturer (example file DEMODS5T.SRC) are painfully slow: a byte read takes 106 processor cycles, a byte write takes 112 cycles, and clock opening takes 1929 cycles. Therefore, access to all real-time-clock registers (open/read or open/write sequences) lasts more than 2800 cycles.

Listing 1 uses a different control scheme; the protocol logic resets only during system start-up, which consumes 436 cycles. Also, the **listing** linearizes the short loops used in the original procedures to open the clock and read/write the data byte. The result is that a byte read takes 51 cycles, a byte write takes 57 cycles, a whole real-time-clock read takes 926 cycles,

and a real-time-clock write takes 972 cycles. The potential drawback of this option, with respect to the original approach, is that interrupt-service routines executed during real-time-clock access must not address external data memory, because any MOVX would interfere with the clock-access protocol.

You can download **Listing 1** and other related listings from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2187. (DI #2187) e

To Vote For This Design, Circle No. 415

LISTING 1—REAL-TIME-CLOCK-ACCESS ROUTINE

```
;*****************************************
; time-optimized RTC access routines for DS5000T
; based on DEMODS5T.SRC of Dallas Semicondictor
; JCh - January 1998
;*****************************************
$MOD5000

;*****************************************
; Close the RTC in case it was open before
;*****************************************
CloseRTC:
        push    MCON            ; save MCON register
        orl     MCON,#4         ; switch to CE2
        mov     DPTR,#0004      ; set up for data input
        movx    A,@DPTR         ; reset pattern detector
        call    RBYTE_          ; read 1st byte
        call    RBYTE_          ; read 2nd byte
        call    RBYTE_          ; read 3rd byte
        call    RBYTE_          ; read 4th byte
        call    RBYTE_          ; read 5th byte
        call    RBYTE_          ; read 6th byte
        call    RBYTE_          ; read 7th byte
        call    RBYTE_          ; read 8th byte
        pop     MCON            ; restore MCON register
        ret                     ; return

;*****************************************
; Read all RTC registers into predefined memory locations
;*****************************************
ReadRTC:
        push    MCON            ; save MCON register
        orl     MCON,#4         ; switch to CE2
        mov     DPTR,#0004      ; set up for data input
        movx    A,@DPTR         ; reset pattern detector
        mov     A,#0C5H         ; load 1st byte of pattern
        call    WBYTE_          ;
        cpl     A               ; generate 2nd pattern byte
        call    WBYTE_          ;
        swap    A               ; generate 3rd pattern byte
        call    WBYTE_          ;
        cpl     A               ; generate 4th pattern byte
        call    WBYTE_          ;
        swap    A               ; generate 5th pattern byte
        call    WBYTE_          ;
        cpl     A               ; generate 6th pattern byte
        call    WBYTE_          ;
        swap    A               ; generate 7th pattern byte
        call    WBYTE_          ;
        cpl     A               ; generate 8th pattern byte
        call    WBYTE_          ;
        mov     DPL,#4          ; set up for data input
        call    RBYTE_          ; read a byte
        mov     RTChdr,A
        call    RBYTE_          ; read a byte
        mov     RTCsec,A
        call    RBYTE_          ; read a byte
        mov     RTCmin,A

        call    RBYTE_          ; read a byte
        mov     RTChrs,A
        call    RBYTE_          ; read a byte
        mov     RTCdow,A
        call    RBYTE_          ; read a byte
        mov     RTCday,A
        call    RBYTE_          ; read a byte
        mov     RTCmon,A
        call    RBYTE_          ; read a byte
        mov     RTCyrs,A
        pop     MCON            ; restore MCON register
        ret

;*****************************************
;oad all RTC registers from predefined memory locations
;*****************************************
teRTC:
        push    MCON            ; save MCON register
        orl     MCON,#4         ; switch to CE2
        mov     DPTR,#0004      ; set up for data input
        movx    A,@DPTR         ; reset pattern detector
        mov     A,#0C5H         ; load 1st byte of pattern
        call    WBYTE_          ;
        cpl     A               ; generate 2nd pattern byte
        call    WBYTE_          ;
        swap    A               ; generate 3rd pattern byte
        call    WBYTE_          ;
        cpl     A               ; generate 4th pattern byte
        call    WBYTE_          ;
        swap    A               ; generate 5th pattern byte
        call    WBYTE_          ;
        cpl     A               ; generate 6th pattern byte
        call    WBYTE_          ;
        swap    A               ; generate 7th pattern byte
        call    WBYTE_          ;
        cpl     A               ; generate 8th pattern byte
        call    WBYTE_          ;
        mov     A,RTChdr
        call    WBYTE_          ; write a byte
        mov     A,RTCsec
        call    WBYTE_          ; write a byte
        mov     A,RTCmin
        call    WBYTE_          ; write a byte
        mov     A,RTChrs
        call    WBYTE_          ; write a byte
        mov     A,RTCdow
        call    WBYTE_          ; write a byte
        mov     A,RTCday
        call    WBYTE_          ; write a byte
        mov     A,RTCmon
        call    WBYTE_          ; write a byte
        mov     A,RTCyrs
        call    WBYTE_          ; write a byte
        pop     MCON            ; restore MCON register
        ret                     ; return

;*****************************************
; Read one byte from the RTC and return it in ACC
;*****************************************
RBYTE_:
        movx    A,@DPTR         ; input 1st bit
        mov     C,ACC.7         ; move it to carry
        mov     B.0,C           ; save the data bit
        movx    A,@DPTR         ; input 2nd bit
        mov     C,ACC.7         ;
        mov     B.1,C           ;
        movx    A,@DPTR         ; input 3rd bit
        mov     C,ACC.7         ;
        mov     B.2,C           ;
        movx    A,@DPTR         ; input 4th bit
        mov     C,ACC.7         ;
        mov     B.3,C           ;
        movx    A,@DPTR         ; input 5th bit
        mov     C,ACC.7         ;
        mov     B.4,C           ;
        movx    A,@DPTR         ; input 6th bit
        mov     C,ACC.7         ;
        mov     B.5,C           ;
        movx    A,@DPTR         ; input 7th bit
        mov     C,ACC.7         ;
        mov     B.6,C           ;
        movx    A,@DPTR         ; input 8th bit
        mov     C,ACC.7         ;
        mov     B.7,C           ;
        mov     A,B             ; copy result
        ret                     ; return

;*****************************************
; Write one byte from the accumulator to the RTC
;*****************************************
WBYTE_:
        mov     B,A             ; save the accumulator
        anl     A,#1            ; set up bit for output
        mov     DPL,A           ; set address to write bit
        movx    A,@DPTR         ; output 1st bit
        clr     A               ;
        mov     C,B.1           ;
        rlc     A               ;
        mov     DPL,A           ;
        movx    A,@DPTR         ; output 2nd bit
        clr     A               ;
        mov     C,B.2           ;
        rlc     A               ;
        mov     DPL,A           ;
        movx    A,@DPTR         ; output 3rd bit
        clr     A               ;
        mov     C,B.3           ;
        rlc     A               ;
        mov     DPL,A           ;
        movx    A,@DPTR         ; output 4th bit
        clr     A               ;
        mov     C,B.4           ;
        rlc     A               ;
        mov     DPL,A           ;
        movx    A,@DPTR         ; output 5th bit
        clr     A               ;
        mov     C,B.5           ;
        rlc     A               ;
        mov     DPL,A           ;
        movx    A,@DPTR         ; output 6th bit
        clr     A               ;
        mov     C,B.6           ;
        rlc     A               ;
        mov     DPL,A           ;
        movx    A,@DPTR         ; output 7th bit
        clr     A               ;
        mov     C,B.7           ;
        rlc     A               ;
        mov     DPL,A           ;
        movx    A,@DPTR         ; output 8th bit
        mov     A,B             ; restore the accumulator
        ret                     ; return


        DSEG
        ORG     30H

RTChdr: DS      1               ; 00-99 0.01 seconds
RTCsec: DS      1               ; 00-59 seconds
RTCmin: DS      1               ; 00-59 minutes
RTChrs: DS      1               ; 00-23 or 01-12 hours (b7 12/24 b5
AM/PM)
RTCdow: DS      1               ; 01-07 weekday (b5 -OSC)
RTCday: DS      1               ; 01-31 day
RTCmon: DS      1               ; 01-12 month
RTCyrs: DS      1               ; 00-99 year

        END
```

Limiting amplifier is digitally programmable

V MANOHARAN, NAVAL PHYSICAL AND OCEANOGRAPHIC LABORATORY, KOCHI, INDIA

Amplitude limiters are necessary in many systems, such as radar and FM receivers, for which the system cannot allow the amplitude of the signal to exceed the given positive, negative, or both limits. In the circuit in **Figure 1**, amplifier IC_{4B}'s maximum output is digitally programmable over ±2 to ±10V in 2^n steps, where n is the number of bits of the DAC. IC_1, a precision 10V reference, provides a full-scale reference current, $I_{REF}=V_{REF}/R_1=2$ mA, to IC_2, a multiplying DAC.

IC_3's output voltage, V_L, is the sum of the product of the digital word and unipolar reference voltage and IC_1's dc offset as follows:

$$V_L=\left(\frac{V_{REF}}{R_1}\bullet\frac{N}{2^n}\bullet R_2\right)+\frac{R_4}{R_3+R_4}\bullet V_{REF},$$

where N can assume values of 0 to 2^n–1.

When all digital inputs are set to a logic low, N=0,

$$V_{L(MIN)}=\frac{R_4}{R_3+R_4}\bullet V_{REF}.$$

For the values of R_3, R_4, and V_{REF} in this example, $V_{L(MIN)}$=1V. When all the digital inputs are set to logic high (n=8, and N=255),

$$V_{L(MAX)}=\frac{V_{REF}}{R_1}\bullet\frac{2^n-1}{2^n}\bullet R_2+\frac{R_4}{R_3+R_4}\bullet V_{REF}$$
$$=\frac{V_{REF}}{R_1}\bullet\frac{255}{256}\bullet R_2+1V.$$

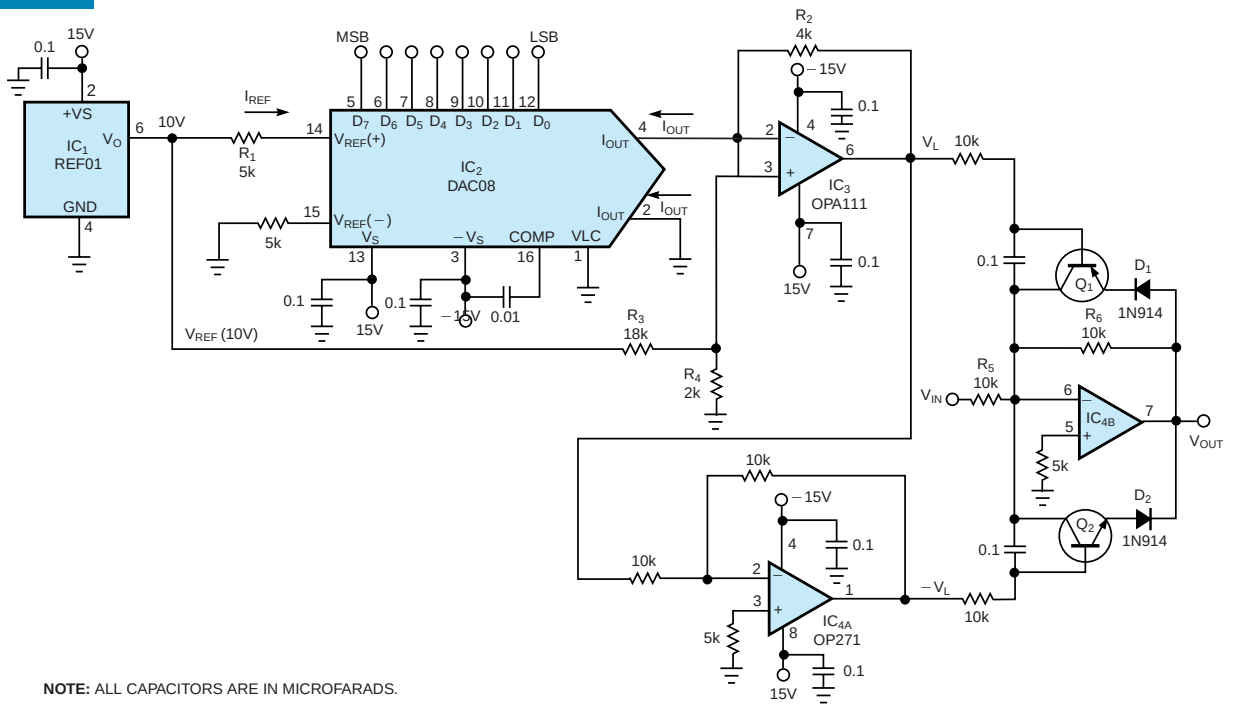
For the values of R_1 and R_2, $V_{L(MAX)}\approx$9V.

Within the limiting levels, the amplifier does not modify its input signal but provides a gain of $A_V=-R_6/R_5$. As V_{OUT} rises above V_L+1V—adding 1V overcomes the potential drops of the base-emitter junctions of Q_1 and D_1—the base-emitter junction of Q_1 becomes forward-biased, allowing the collector current to flow to the summing node, thus limiting V_{OUT}. A similar action occurs with Q_2 and D_2 as V_{OUT} goes below $-V_L$–1V.

You can thus program the limiting levels or the maximum output voltage of the amplifier symmetrically over $V_{L(MIN)}$+1V to $V_{L(MAX)}$+1V with a resolution of $[(V_{L(MAX)}+1)-(V_{L(MIN)}+1V)]/2^n$V in accordance with the 8-bit digital-input binary word. The circuit becomes a programmable positive/negative limiting amplifier if you remove the appropriate diode-transistor pairs from the feedback. (DI #2201) e

To Vote For This Design, Circle No. 416

FIGURE 1



The maximum output of this amplitude limiter is digitally programmable over ±2 to ±10V in 2^n steps, where n is the number of bits of the DAC.

Get more than three colors from a dot-matrix LED

W KURDTHONGMEE, NAKORN SI THAMMARAT, THAILAND

Dot-matrix LEDs find wide use in advertising displays. Products now on the market range from an inexpensive 5×8 (row-by-column) single-color LED to an expensive 8×8 RGB device. The method provided here allows you to obtain more than three main colors from an 8×8 tricolor LED. In fact, tricolor dot-matrix LEDs have only two LED dies—red and green. When you apply current to one, you obtain a red or a green color. When you apply current to both, orange results. The circuit in **Figure 1**, used in conjunction with the MCS-51 code in **Listing 1**, works efficiently in controlling the LED to generate various shades of the three colors.

To add tones or shades of the main colors to the tricolor LED, you do not need to modify the circuit in **Figure 1**; you need only consider the software. Software modifications consist of adding more color planes or pages of display buffer, adding memory locations (mapped onto the LED dots), and increasing the number of refresh times, in which the controller updates all LED dots to cover all added color planes.

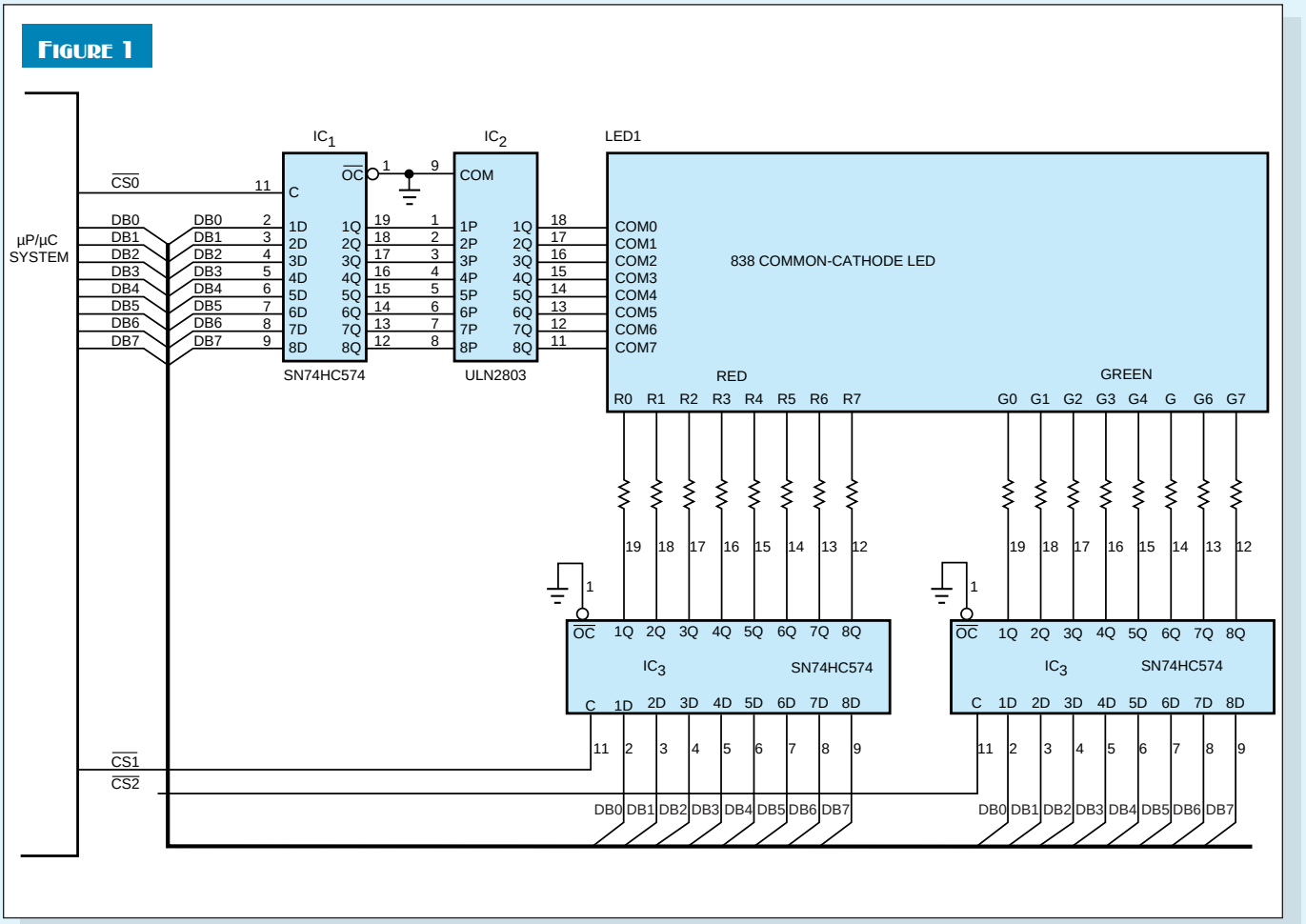
TABLE 1—VALUE IN COLOR PLANE MAPPED TO DOT I

Red 1	Red 2	Green 1	Green 2	Color
0(1)	1(0)	0	0	Red 50%
1	1	0	0	Red 100%
0	0	0(1)	1(0)	Green 50%
0	0	1	1	Green 100%
0(1)	1(0)	0(1)	1(0)	Orange 50%
1	1	1	1	Orange 100%
0	0	0	0	Blank

For example, if you decide to use four color planes, divided into two red and two green planes, for dot i of the dot-matrix LED, you'll obtain the shades listed in **Table 1**.

In addition, by allocating eight color planes (four red and

FIGURE 1



A few TTL circuits and some MCS-51 code allow you to obtain more than three colors from a tricolor dot-matrix LED.

four green), you can obtain the color shades listed in **Table 2**. Note that only the number of ones in the color planes controls color appearance. Therefore, the permutations do not change the color, as long as the numbers of ones in **Table 2** remain constant. For example, the values 0110, 1001, 1100, and 0011 for R1 through R4 all produce the same color: orange 50%. You can download **Listing 1**—as well as the MCS-51 code that produces 13 colors from an 8×8 tricolor LED—from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go to the "Software Center" to download the file from DI-SIG #2195.

Note that, in practice, bytewide output ports control the LED. To assign a color to a dot, the routine must extract a bit from a byte and then assign the bit value of the selected color plane by plane. (DI #2195) e

To Vote For This Design, Circle No. 417

TABLE 2—VALUE IN COLOR PLANE MAPPED TO DOT 1

Red 1	Red 2	Red 3	Red 4	Green 1	Green 2	Green 3	Green 4	Color
0	0	0	1	0	0	0	0	Red 25%
0	0	1	1	0	0	0	0	Red 50%
0	1	1	1	0	0	0	0	Red 75%
1	1	1	1	0	0	0	0	Red 100%
0	0	0	0	0	0	0	1	Green 25%
0	0	0	0	0	0	1	1	Green 50%
0	0	0	0	0	1	1	1	Green 75%
0	0	0	0	1	1	1	1	Green 100%
0	0	0	1	0	0	0	1	Orange 25%
0	0	1	1	0	0	1	1	Orange 50%
0	1	1	1	0	1	1	1	Orange 75%
1	1	1	1	1	1	1	1	Orange 100%
0	0	0	0	0	0	0	0	Blank

LISTING 1—MCS-51 CODE FOR SIX COLORS FROM A TRICOLOR LED

```
;*****************************************************************************
; Program Name : TriCol.asm
; Generate three colours from triple colours dot matrix LED.
;*****************************************************************************
; Note:
; Only a sample implementation on 8x8 triple colours LED
;*****************************************************************************
; Author       : Wattanapong Kurdthongmee,
;              : School of Physics, Walailak University, Thaiburi,
;              : Tha-Sa-La, Nakorn si Thammarat, 80160 Thailand.
;*****************************************************************************

;*****************************************************************************
; Global variable declarations:
;*****************************************************************************
temp    equ     10h
dBuf    equ     11h

;*****************************************************************************
; Interrupt vectors
;*****************************************************************************
        org     0000h
        jmp     main
        org     000bh
        jmp     Timer0SR

;*****************************************************************************
; Timer 0 interrupt service routine
; Update row-by-row the triple-colour 8x8 dot matrix LED.
; To make more tones of colour, this routine reads 4 set of display buffer
;  and scan 4 times.
;*****************************************************************************
Timer0SR:
        setb    rs0             ; Select set of registers in bank 1
        push    dph
        push    dpl
        push    psw
        push    acc
         mov    th0,#0ffh       ; Re-initialize timer registers
         mov    tl0,#00h
                                ;----------------------------------------
        mov     dptr,#0a000h    ; Clear enable controlled port
        mov     a,#00h          ;  before updating
        movx    @dptr,a
                                ;----------------------------------------
        mov     a,#dBuf
        add     a,r0
        push    acc
        mov     r1,a
        mov     a,@r1           ;----------------------------------------
                                ; Read from current address at a current

                                ;  plane and update column controlled port
                                ;  colour by colour
        mov     dptr,#8000h     ; Red colour port
        movx    @dptr,a
        pop     acc
        add     a,#08h          ; Difference display buffer between RED and
                                ;  green colour plane
        mov     r1,a
        mov     a,@r1
        mov     dptr,#9000h     ; Green colour port
        movx    @dptr,a
                                ;----------------------------------------
        mov     dptr,#0a000h    ; Enable current row (controlled by ULN2803
        mov     a,r3
        movx    @dptr,a
        rl      a
        mov     r3,a
                                ;----------------------------------------
        inc     r0              ; Next row (there are 8-row for 8x8 LED)
        cjne    r0,#08h,T0SR_r
        mov     r0,#00h
        mov     r3,#01h
T0SR_r:         pop     acc             ; Restore registers
        pop     psw
        pop     dpl
        pop     dph
        clr     rs0
        reti

;#############################################################################
; Main program starting here
;#############################################################################
;*****************************************************************************
main:   mov     sp,#60h

        setb    rs0             ; Initialise register in bank 1 to be used
        mov     r0,#00h         ;  the timer interrupt service routine.
        mov     r3,#01h
        clr     rs0
         mov    tmod,#21h       ; timer 0: mode 1, timer 1: mode 2
         mov    tcon,#0ddh
         mov    th0,#0fdh       ; Initial value for timer
         mov    tl0,#00h
         setb   ea              ; Enable all interrupts
         setb   tr0             ; Start timer
         setb   et0             ; Enable timer interrupt 0

                                ; Sample patterns to show different shades
                                ;  colours on the LED.
        mov     r0,#dBuf
        mov     r1,#00h


main_0:         mov     a,#01010011b
        mov     @r0,a
        inc     r0
        inc     r1
        cjne    r1,#08h,main_0
        mov     r0,#dBuf+8
        mov     r1,#00h
main_1:         mov     a,#10101111b
        mov     @r0,a
        inc     r0
        inc     r1
        cjne    r1,#08h,main_1
        sjmp    $

        end
```

Implement a nine-data-bit UART on a PC

AUBREY KAGAN, WEIDMULLER LTD, MARKHAM, ON, CANADA

Many µCs, such as the 8051 and the 68HC11, can support a ninth data bit on the asynchronous serial port. This bit is useful in multidrop applications in which you can use it to denote an address on the serial bus, as opposed to data destined for a particular address. The UART used in IBM PCs (and clones) does not directly support this operating mode. However, through some software manipulation, you can add the PC to a serial bus and integrate it into a ninth-bit system, albeit with some limitations.

The method differs for data reception and transmission. As a result, the PC can work only in half-duplex mode. Because half-duplex communication is common practice on PC networks, this limitation is not a significant drawback. The technique also requires that the CPU check each incoming byte for the ninth bit. (You can usually configure a µC to generate an interrupt when the ninth bit is set.) For the PC to receive the nine bits, it is necessary to treat the ninth bit as a parity bit. Although it's impossible to read the parity bit in the PC's UART directly, it is possible to analyze the received data byte and determine what the parity should be.

If analysis reveals a parity error, then the value of the ninth bit is opposite to the calculated parity. If no error exists, then the value of the ninth bit is equal to the calculated parity. In the 16550 UART, the FIFO includes the three error bits with each data byte, so the parity error (or lack thereof) is always associated with the current data byte. It is possible, however, to disable the FIFO feature. The technique for transmission is slightly different. The 8250/16450/16550 UART has a forced-parity format (also known as a "stick" parity), in which you can set the parity to a one or to a zero. You do this by setting bit 5 (stick parity) and bit 3 (parity enable) in the UART's line-control register (LCR). The transmitted parity bit is then the logical inverse of bit 4 of the LCR.

In the sample code in **Listing 1**, address 0xff (with bit 9 set) is reserved and used to indicate the last byte of the transmission. The first byte of the transmission is an address, and it transmits with bit 9 set. The RS-232C port connects to an RS-232C/RS-485 converter, where the RTS line controls the direction. The code given here is not interrupt-driven, but you could implement it as an interrupt-driven routine. The code comprises three modules: background (back.cpp), serial procedures (serial.cpp), and memory declaration (mem.cpp). Note that mem.cpp declares one include file (mem.h) for the public memory. You can download the files from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go to the Software Center to download the files from DI-SIG #2198. (DI #2198) e

To Vote For This Design, Circle No. 418

LISTING 1—BACKGROUND CODE FOR NINTH-BIT TRANSMISSION

```
//background program
//developed with Turbo C++
//operating under DOS

#include "mem.h"
#include <conio.h>

#define RTS 0x2

//prototypes
void setupUART (void);
void deAssert (int ControlPin);
void Assert (int ControlPin);
unsigned char SerialIn(void);
void UARTrx(void);
void UARTtx(void);
unsigned char UART_TX_clear( void);
unsigned char SerialOut (void);
unsigned int checksum (unsigned char NumberOfBytes);

void main (void)
{
unsigned int j;

     comport=1;
     //setting to COM1

     module_address=0xa;
     //PC address=10 decimal

     //other transmission constants
     setupUART();
     //initialise the UART

     capture_enabled=0;
     rx_pnt=0;
     //initialise variables

     Assert(RTS);
     //turn the RS485 buffer to receive

     while (1)
     {
          /*the actions are divided into several states as indicated by
the variable "phase".
          Phase=0- waiting for a complete serial message
          Phase=1- preparing a response
          Phase=2-wait for end of transmission
          Phase=3- wait for message to completely clear the UART (buffers
empty) & then
               re-enable reception (turn RS485 buffer around)*/

     switch (phase)
     {
          case 0:
               if (SerialIn())
               //checking for complete message received
               {
                    //now to process the input
                    phase++;
                    UARTtx();
                    //prepare UART to send
               }
               break;
          case 1:
               //prepare to transmit
               rx_buff[0]=0x0;     //destination address
               rx_buff[1]=0x13;    //response
               rx_buff[2]=0xff;
               //set last byte.

               number_of_characters=3;
               //variable for transmit routine
               rx_pnt=0;
               //intitialise the fetch pointer
               phase++;
               break;
          case 2:
               if (SerialOut())
               {//at the end of the message
                //bump on to wait for complete transmission
                    phase++;
               }
               break;
          case 3:
               //wait for message to clear
               if (UART_TX_clear())
               {
                    UARTrx();
                    phase=0;
               }
               break;
          default:
               break;
     }

     if (kbhit())
      {//terminate execution if any key pressed.
         break;
      }
}
```

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____ Fax _____

E-mail _____

My e-mail address may be published Yes____ No____

Company _____

_____

Address _____

_____

Country _____ ZIP _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

Boost converter generates three analog rails

TIM HERKLOTS, MAXIM INTEGRATED PRODUCTS LTD, THEALE, BERKSHIRE, UK

The standard boost converter in **Figure 1** uses not only IC_1, C_1, L_1, D_1, and C_2 to generate a main 5V output, but also additional small, low-cost components to provide two auxiliary supply rails of 10 and –5V. These auxiliary outputs are useful for analog circuitry in small handheld instruments, which often require supply voltages greater than the signal range. Input voltages of 0.8 to 5.5V, which is equivalent to voltages from a battery pack of one to three cells, sustain the main regulated output of 5V±2%. With an input of 1.8V from two flat cells, for instance, and with the other rails unloaded, the circuit can produce 25 mA with 80 to 90% efficiency.

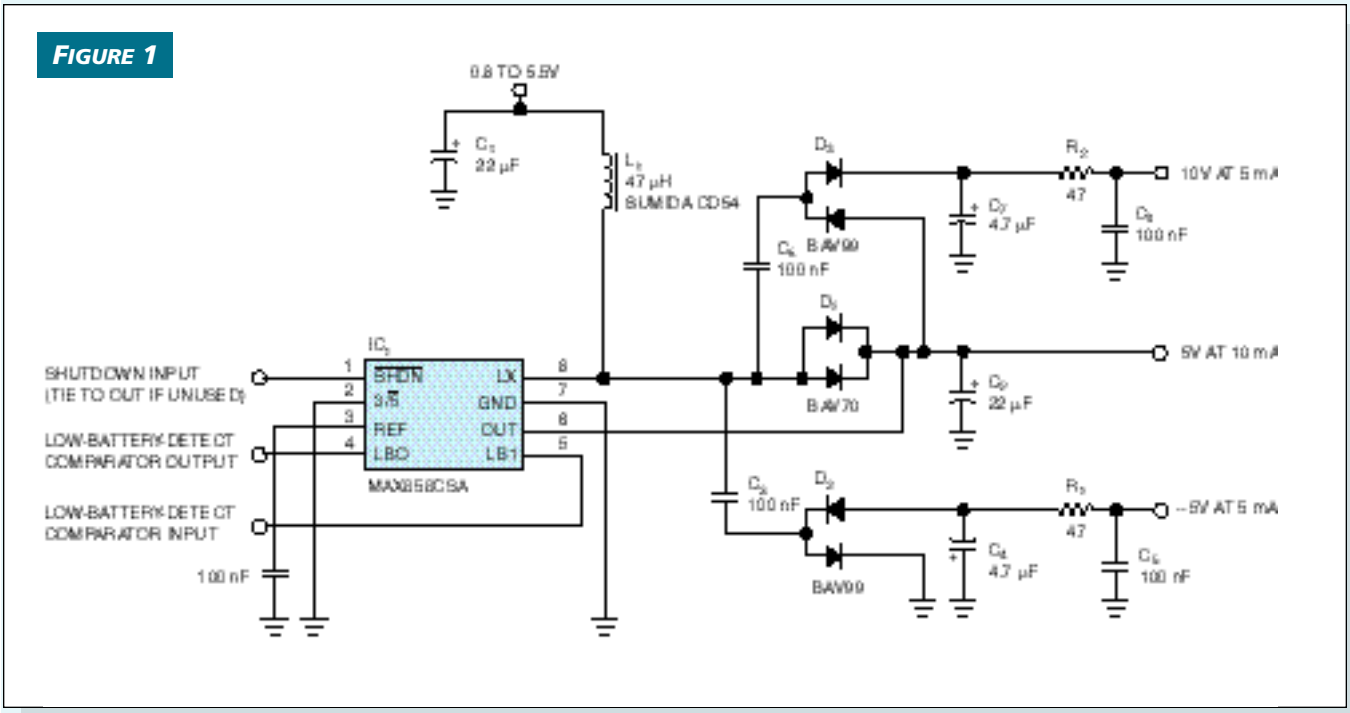
The converter's LX switching node drives low-cost, discrete charge pumps via "flying capacitors" C_3 and C_6 to create the –5V and 10V outputs. The LX node switches between 0V and a level-one diode drop above the 5V rail, so the charge pumps' drive voltage is reasonably well-regulated. Moreover, the drop across D_1 roughly compensates for diode drops in the two charge-pump outputs. IC_1's internal control scheme also assists in regulating the auxiliary outputs. This IC's current-limited, minimum-off-time, pulse-frequency modulation constantly adapts its switching frequency to the net load current; the frequency increases when the load increases, producing a greater transfer of energy via the flying capacitors. The result is a type of pseudoregulation for the charge-pump outputs.

These analog supply rails can drive precision op amps, such as the MAX400 and OP-07, whose input common-mode-rejection and output-range specifications are 2 to 3V within the supply rails. Thus, the rails are good enough if the –5V output is less than –3V and the 10V output is more than 8V. Accordingly, the component choices in **Figure 1**, such as the lossy RC output filters and silicon signal diodes in place of Schottky diodes, provide for minimal cost and ripple rather than maximum regulation. The 4.7-µF capacitors, C_4 and C_7, can be high-ESR, commodity, multilayer-ceramic types with 16V ratings, a 1206 case, and a Y5V dielectric, such as the 1206YG475ZAT2A from AVX Corp (www.avxcorp.com).

The output ripple varies with the supply voltage and output load. Operating with an input voltage of 1.8V, the circuit produces ripple amplitudes over the load of 2 to 10 mV p-p for the 10V rail and 15 to 30 mV p-p for the –5V rail. By increasing C_5 and C_8 to 2.2 µF, you can reduce these ripple levels to 1 and 5 mV, respectively.

With no load on the auxiliary rails, the 5V output's maximum available load current rises with input supply voltage (**Figure 2a**). You can increase this available output power by replacing D_1 with a lower loss Schottky diode. At an input of 1.8V, the output power available for the three rails (loaded with 10 mA at 5V, 5 mA at 10V, and 5 mA at –5V) is somewhat less than 125 mA; with a 5-mA load, the 10V and –5V outputs are approximately 9.75 and –3.7V, respectively (**Figure 2b**). A 2.7V input based on three flat cells yields

FIGURE 1



Adding external charge pumps to this 5V boost converter produces auxiliary analog rails of 10 and –5V.

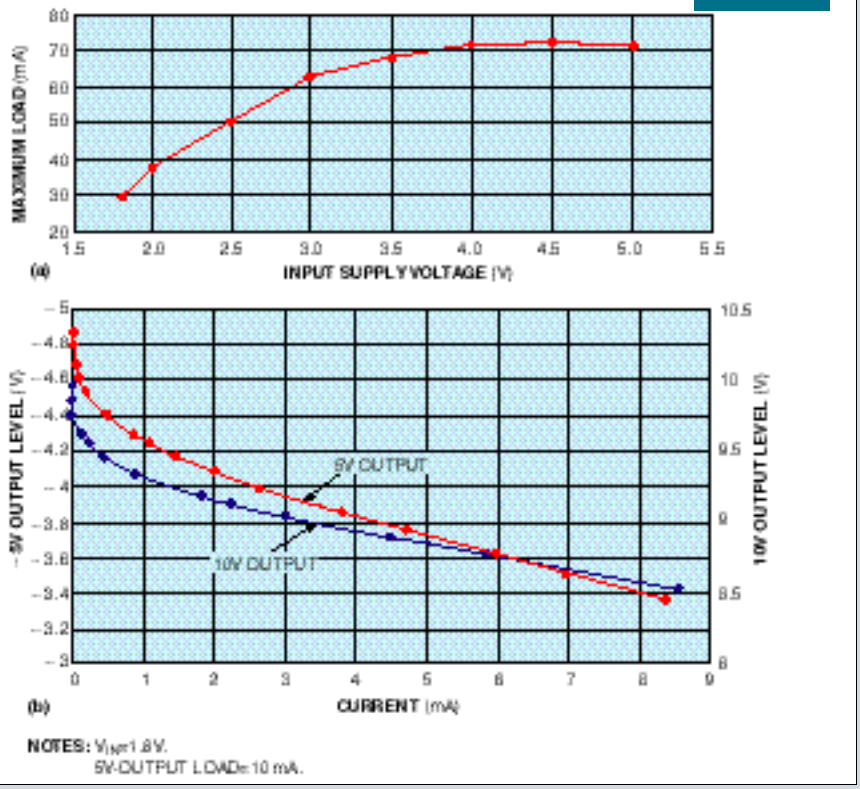
around 275 mW.

The MAX858 operates with peak inductor currents of 125 mA. If you need more current, you can replace this IC with related parts that have 500 mA and 1A ratings. Note that these changes require different passive components; the inductor and main output diode ratings must match the inductor's peak current. The charge pumps can remain the same if their output currents don't change much.

You can also retain the cheap, common, commodity dual diodes D_1, D_2, and D_3, but detail specifications vary, so look carefully at data sheets for the part you actually use. For example, the BAV70's dc forward current, I_F, and peak forward surge current, I_{FSM} for 1 μsec, differ among manufacturers. For the Motorola (www.motorola.com) part, I_F=200 mA, and I_{FSM}=500 mA. For National Semiconductor (www.national.com), I_F=600 mA, and I_{FSM}=2A. For Philips (www.philips.com), I_F=125 mA, and I_{FSM}=4A, and for Vishay-Siliconix (www.siliconix.com), I_F=250 mA, and I_{FSM}=4.5A. This caution is advisable in all second-source considerations. (DI #2200) EDN

To Vote For This Design, Circle No. 514

FIGURE 2



With auxiliary rails unloaded, the 5V output's maximum available load current rises with input supply voltage (a). The auxiliary-output voltage levels depend on the load current (b).

Automatic-exposure scheme uses CCD shutter

GIOVANNI ROMEO AND SANDRO RAO, ISTITUTO NAZIONALE DI GEOFISICA, ROME, ITALY

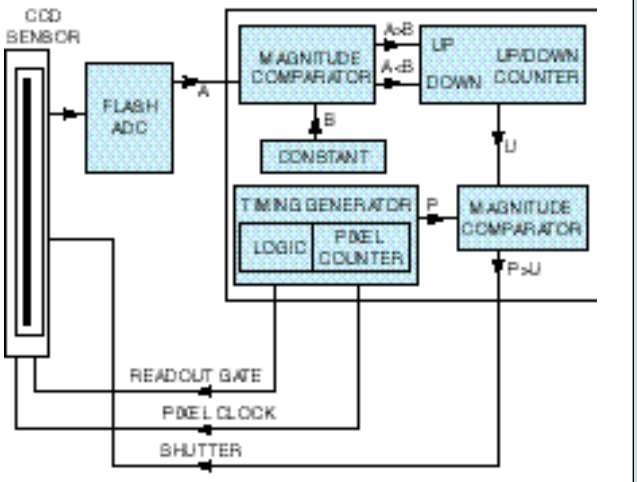
This application follows the Design Idea, "Peak detector maximizes CCD-sensor range" (*EDN*, Aug 15, 1996). Its aim was to optimize the performance of an A/D converter used to digitize a linear CCD sensor's analog output. The method involved stretching the upper reference of the flash ADC for the highest lit pixel in the array. The method works well, but does not obtain the best performance from the CCD, which can saturate for overexposure or can produce noise for underexposure. **Figure 1** shows a better method that you can use with CCD sensors that provide a shutter facility. The shutter signal in a modern CCD array (such as the Sony ILX703A) removes the electrical charges the light produces during the exposure. Thus, the time between the shutter signal and the data reading is the exposure time. The circuit in **Figure 1** simply moves the shutter pulse between two subsequent readout gates.

The circuit digitally compares the ADC's output with the desired level (near the maximum ADC output). If the output exceeds the threshold level, the up/down counter increments; otherwise, it decrements. The magnitude comparator compares the up/down counter's register contents with the pixel-counter data, and, when the data exceeds the contents, the shutter signal activates. The system requires an average settling time of (number of pixels)/(2×pixel time) and, in the steady-state condition, oscillates with a period of one pixel time. Our application required obtaining the shape of the light distribution, neglecting the absolute illumination information. You can use the contents of the

up/down counter's register as a scale factor, when you need to measure the absolute illumination.

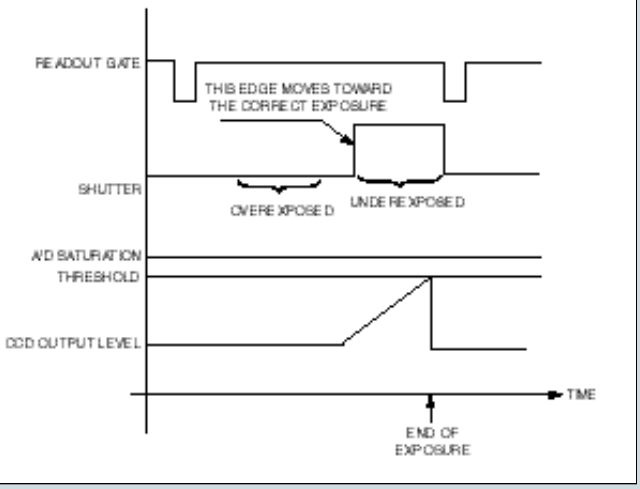
Figure 2 shows the waveforms in the system. A 1016 PLD generates the signals to control the system and the CCD. **Listing 1** gives the Abel program for the 1016. You can download the file from *EDN*'s Web site www.ednmag.com. At the registered-user area, go into the Software Center to download the files from DI-SIG, #2213. You can better understand the CCD's operation by referring to Sony's 1992 application note, "Linear Sensor Application Note." (DI #2213) EDN

FIGURE 1



To adjust the automatic-exposure system, set constant B near the maximum ADC level, tuned to match the maximum unsaturated CCD output.

FIGURE 2



The shutter signal keeps the sensor empty after asserting the readout gate. The exposure begins just in time to keep the CCD's output at a level that uses the ADC's entire dynamic range.

ABEL PROGRAM FOR AUTOMATIC-EXPOSURE PLD

```
MODULE ccd

TITLE ' ccd, timing and shutter generator '
shutter INTERFACE (clk,p11..p0,uD11..uD0,ff -> sh);
  shut  FUNCTIONAL_BLOCK shutter;

clk                     pin;
[AD9..AD0]              pin;  //comparator input
[CK1..CK0]              pin       istype 'reg,buffer';  //frequency
divider.
[P11..P0]               node      istype 'reg,buffer';  // pixel counter
[UD11..UD0]             node      istype 'reg,buffer';  // UP/DOWN
counter
RDG                     pin;        //reg pulse
CCDCLOCK, ADCLOCK       pin       istype 'buffer';      // ccd and AD
clock
OUT                     node;                           // UP/DOWN
selection
FF                      node istype 'reg_D';   //peak detector FF
SHU                     pin; shutter signal

"Data
Input   = [AD9..AD0];
Count1 = [P11..P0];
Count2 = [UD11..UD0];

Equations
shut.clk = clk;
SHU = shut.sh;
shut.[p11..p0] = [P11..P0];
shut.[uD11..uD0] = [UD11..UD0];
shut.ff = FF;
FF.D = OUT;
FF.clk = RDG;
FF.AP  = OUT;
[CK1..CK0].clk = clk;
Count1.clk  = CK1;
Count2.clk = !RDG;
[CK1..CK0] := [CK1..CK0] + 1;

when (Count1 == 2149) then Count1 := 0 else Count1 := Count1 + 1;
when ((Count1 > 2083) & (Count1 < 2148)) then ADCLOCK = 0 else ADCLOCK =
!CK1;
when (Count1 < 2086) then CCDCLOCK = CK1 else CCDCLOCK = 0;
RDG = (Count1 < 2107) # (Count1 >= 2147);

when (Input>= [1,1,1,1,1,1,0,1,0,0,1]) then OUT = 1 else OUT = 0;
when (FF== 1) then {
        when (Count2<2997 ) then Count2 := Count2 + 1 else Count2 := 2997;
}
              else {
        when (Count2>0)     then Count2 := Count2 - 1 else count2 := 0; }


End



MODULE shutter

TITLE 'shutter'
clk                     pin;
[p11..p0]               pin;
[uD11..uD0]             pin;  //UP/DOWN counter
ff                      pin;
sh                      pin  istype 'reg'; //shutter
SHU0,SHU1,SHU2,SHU3     node;

EQUATIONS
sh.clk = clk;
SHU0 = [p11..p7] > [uD11..uD7];
SHU1 = [p11..p7] == [uD11..uD7];
SHU2 = [p7..p0] > [uD7..uD0];
SHU3 = (ff==0)&([uD11..uD0]==0);
sh   := SHU0 # (SHU1 & SHU2) # SHU3;

END
```

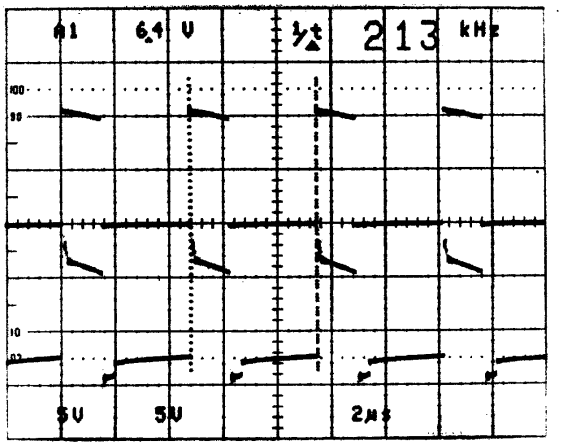
Low-power converter has galvanic isolation

JOSE CARRASCO, UNIVERSIDAD DE VALENCIA, SPAIN

Certain low-power applications require a simple, low-cost, galvanically isolated power supply. **Figure 1** shows a circuit that meets these requirements. The dc/dc converter provides a 12V, 150-mW output using only a few components and a small transformer. The input can come from any power source that supplies 14 to 18V. The CD4049 forms an oscillator that operates at approximately 200 kHz (**Figure 2**). The asymmetry of the oscillator's waveform depends on the value of R. The voltage V_S in **Figure 1** is proportional to the waveform's asymmetry.

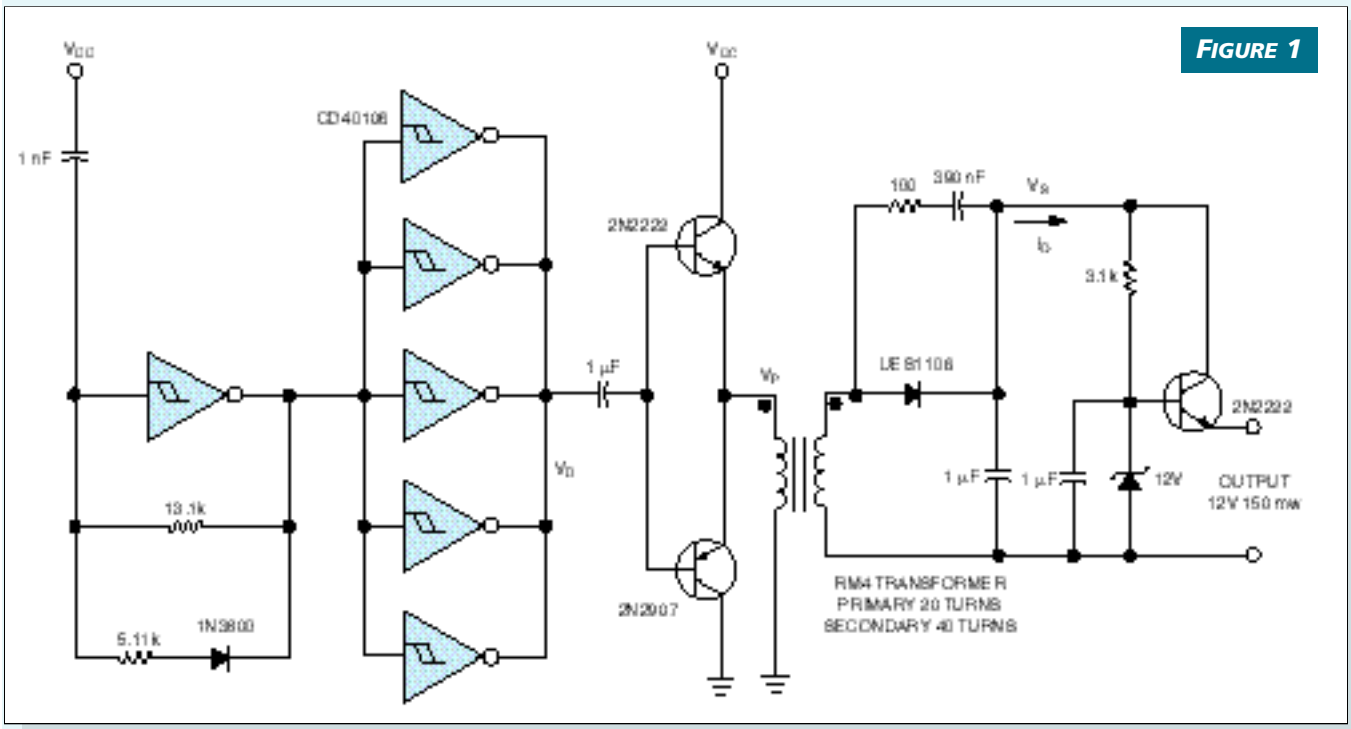
You could also use the circuit as a dc/dc converter with unity transfer ratio by removing the regulator stage at the output. You can easily change the transfer ratio by varying the oscillator's duty cycle (by adjusting R). If you need to increase the output power, remember that in this configuration, the load current flowing through the transformer must be much lower than the magnetizing current. (DI #2214) EDN

FIGURE 2



The symmetry of the waveforms of V_D and V_P in Figure 1 determines the value of V_S.

FIGURE 1



A simple CMOS oscillator, an inexpensive transformer, and a few components form a low-cost, galvanically isolated dc/dc converter

Circuit protects against ac-line disturbances

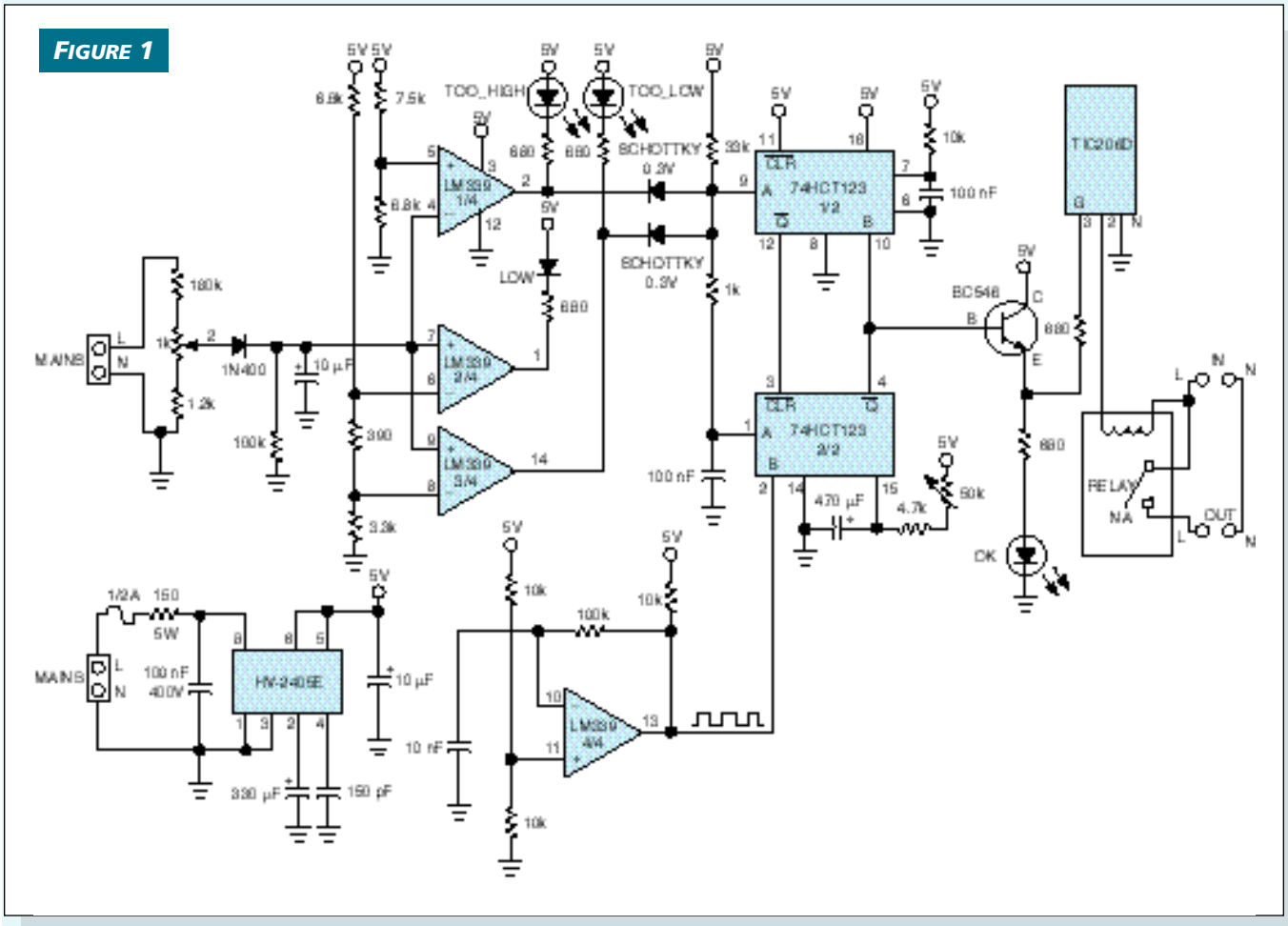
BASILIO SIMOES, ISA, COIMBRA, PORTUGAL

The circuit in **Figure 1** protects the ac line against disturbances. It operates by switching off the power supply upon detection of undervoltage or overvoltage conditions. The circuit thus protects refrigerators, washing machines, air conditioners, and other appliances from permanent damage that could accrue from working outside their specified power requirements. The problem assumes particular importance in underdeveloped countries or regions where the ac-supply network is incorrectly configured, and the voltage frequently drops to levels low enough to damage coils and motors. When the ac-line voltage returns to its nominal level, the circuit automatically resets a switch and reconnects the line voltage.

The input stage contains a voltage divider, which you can adjust with the 1-kΩ potentiometer. The circuit incorporates a rectifying diode and a 10-µF storage capacitor that provides lowpass filtering to stabilize the ac-supply voltage-comparison level. You should adjust the potentiometer such that the normal condition of the ac supply, 220V, corresponds to a 1.97V voltage-comparison level. Three comparison voltages verify the ac-line status, using resistive voltage dividers. The voltages correspond to a 10%-undervoltage warning, a 20%-undervoltage failure level, and a 20%-overvoltage failure level. These comparison voltages correspond to ac-supply voltages of 198, 176, and 264V, respectively. Three sections of the quad open-collector LM339 comparator convert these voltage thresholds to digital signals.

The 10%-undervoltage warning condition turns on a yellow LED. Failure conditions turn on a red LED and trigger

FIGURE 1



Avoid motor burnout, using this circuit that provides undervoltage warning signals and disconnects the line from the load for severe under- and overvoltage conditions.

the dual retriggerable monostable multivibrator, IC_2. The output of the first ,IC_{2A}, is narrow and serves to define a time window that prevents sudden transient disturbances from triggering IC_{2B}. Consequently, if the ac-line voltage quickly returns to its nominal condition, the circuit does not disconnect the load. The output pulse width of the other monostable, which you can adjust via the 50-kΩ potentiometer, defines the time the load remains disconnected after the return of the nominal ac-line voltage.

An RC delay line ensures that when the second monostable triggers, the first one has already activated its Clear input. The fourth comparator of the LM339 produces a high-frequency square wave that continuously retriggers the monostable while the fault condition is present. To save power from the regulated 5V supply and to allow use of this circuit to protect high-current equipment, you should use an output relay whose coil control comes from the power-supply rail. A TIC206D triac, gated by the monostable, switches the relay coil. A green LED indicates that the ac-line level is normal and the relay's contact is closed. IC_1, a Harris HV-2405E offline regulator, supplies the regulated 5V. Because this circuit connects to the ac line, you should use an insulated enclosure, and take care in testing the circuit. (DI #2215)

EDN

To Vote For This Design, Circle No. 517

Fleapower circuit detects short circuits

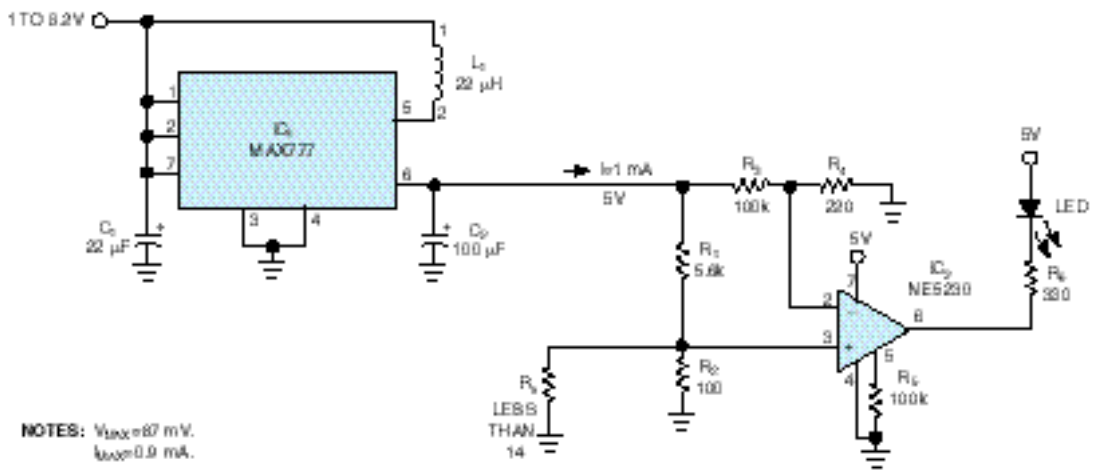
W DIJKSTRA, WAALRE, THE NETHERLANDS

Sometimes, the need arises for a short-circuit tester that supplies a low current to the device under test (DUT) and also uses voltages lower than 100 mV to prevent conduction of semiconductors. The circuit in **Figure 1** meets these requirements. R_1 limits the current in the DUT to 0.9 mA. The voltage on the DUT can not exceed the value set by the ratio $R_2/(R_1+R_2)$. The NE5230 micropower op amp compares the voltage on R_X (representing the DUT) with the voltage at the junction of R_3 and R_4. You can adjust the op amp's supply current by trimming R_5; in this circuit, the current is 0.1 mA. If the value of R_X falls below 14Ω, the output of the op amp switches low and the LED illuminates. The circuit derives its power from a 1.5V battery. IC_1 converts the battery voltage to 5V. (DI #2216)

EDN

To Vote For This Design, Circle No. 518

FIGURE 1



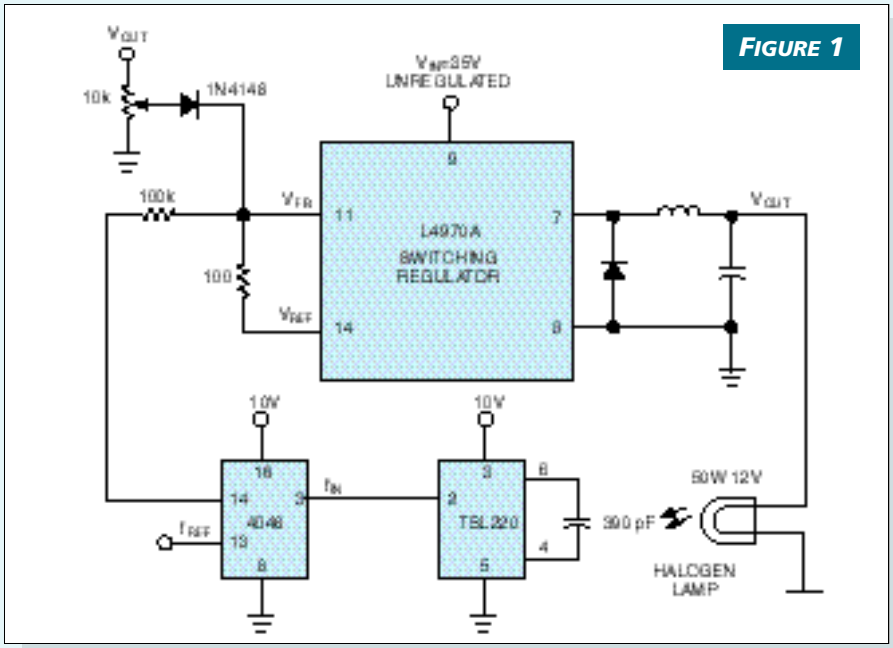
This short-circuit detector uses little power, and provides low currents and voltages to avoid damage to the device under

PLL-based converter controls light source

MASSIMO GOTTARDI, ITC-IRST, TRENTO, ITALY

Using the circuit in **Figure 1** , you can digitally control the light intensity of a lamp. The control loop is based on a PLL, in which the VCO comprises a light-to-frequency converter (TSL220) coupled to a light source that derives its drive from a switching regulator (L4970A). The output of the phase/frequency comparator (4046) serves as the control voltage for the switching regulator. The control voltage is proportional to the frequency error between the reference frequency (f_{REF}) and the signal frequency (f_{IN}) coming from the light-to-frequency converter.

Changing the reference frequency regulates the voltage supplied to the lamp to force the output of the TSL220 to lock to f_{REF}. The two resistors at the output of the 4046 provide an attenuation of 1000 to guarantee the loop stability. As an example, we used the L4970A to drive a 12V, 50W halogen lamp. The control loop operates over a frequency of dc to 500 kHz. To prevent the system from entering a positive-feedback condition, the maximum allowable value of f_{REF} should not exceed the saturation frequency of the TSL220. This maximum value depends on the integrating capacitor used for the light-to-frequency converter and must not exceed 750 kHz. To prevent lamp damage, the 10-kΩ trimmer limits the voltage V_{OUT} applied to the light source. (DI #2219)



A PLL and a light-to-frequency converter allow you to digitally control the intensity of a lamp.

EDN

To Vote For This Design, Circle No. 519

Relay driver saves substantial power

TIM HERKLOTS, MAXIM INTEGRATED PRODUCTS, SOUTHAMPTON, UK

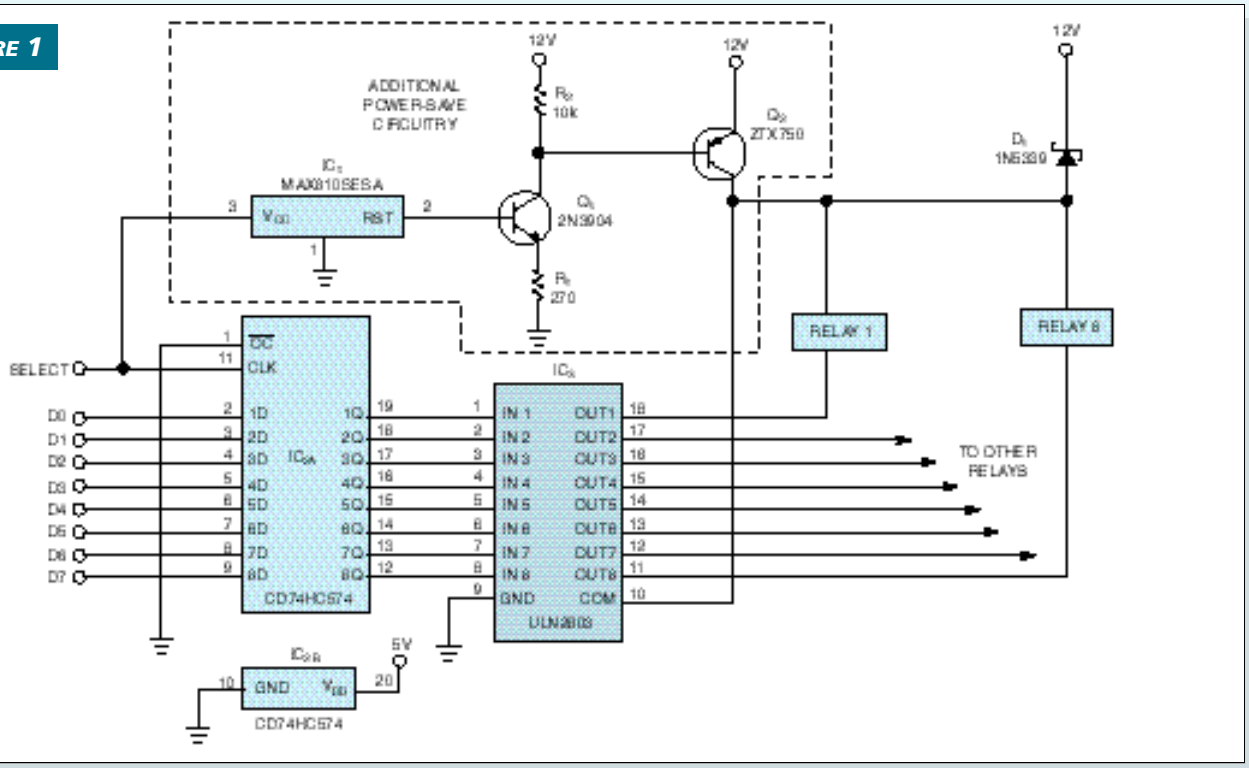
It is common practice to operate relays and solenoids at a reduced holding power once the mechanical actuation takes place. Relays are usually specified to pull in within 3 msec at 80% of the rated voltage and to release at 30% of the rated voltage. The circuit in **Figure 1** drives as many as eight 12V (120Ω coil) power relays, which memory-map into an 8-bit µP bus. An octal latch stores the relay status, where each bit of the 8-bit word serves a separate relay (0=off, 1=on). The latch's Select line latches data on the rising edge. Whenever the relay's status data changes, the relay's drive voltage rises to the full 12V for 140 msec to ensure that the relay pulls in. A series zener diode then reduces the relay's drive voltage by 50% to reduce dissipation.

A ULN2803, an octal Darlington array with base resistors for direct logic interface, drives the relays. A useful feature is the inclusion of eight inductive-load clamping diodes, internally connected between the Outx pins and the Com pin. Com thus connects to the relay-supply rail. The power-saving timing comes from IC_1, a micropower MAX810 processor supervisor powered by the normally high Select line. When the system processor writes to the IC_2 latch, the supply to IC_1 toggles for 200 nsec, causing IC_1 to take its RST output high for 140 to 560 msec. Q_1 operates as a gated current source, dragging current from Q_2, thereby shorting out D_1, a 5.6V zener diode. Hence, the relays receive full bus power during the switching phase. After this period, Q_2 turns off, and D_1 drops the relay supply to the holding voltage. (DI #2217)

EDN

To Vote For This Design, Circle No. 520

FIGURE 1



This power-saving circuit takes advantage of the large turn-on/turn-off hysteresis in electromechanical relays and solenoids.

Step-up/step-down converter takes 2 to 16V inputs

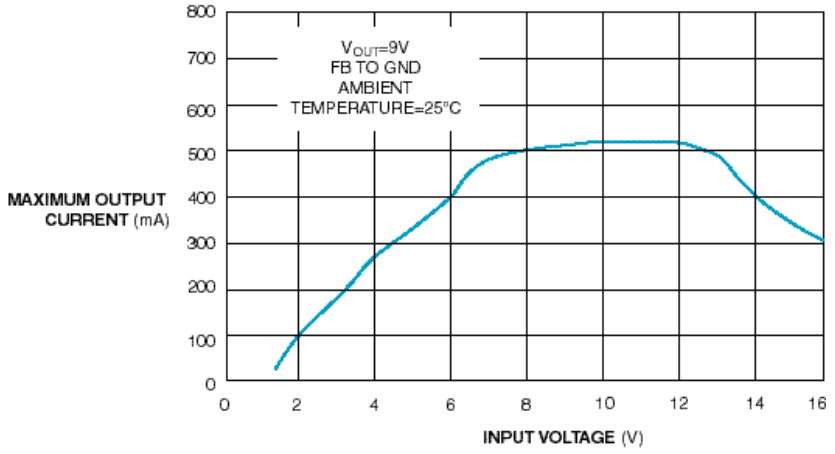
LUCIANO BORDOGNA AND LUCA VASALLI, MAXIM INTEGRATED PRODUCTS, MILAN, ITALY

The circuit in **Figure 1** is a low-cost step-up/step-down dc/dc converter. By definition, its input can range above and below the regulated voltage. The circuit includes a simple switch-mode boost converter (IC_1) that contains a comparator normally used to detect low battery voltage. In this case, the comparator controls an external pnp transistor that operates as a linear regulator. IC_1 steps up V_{IN} (2V minimum) to the level of V_X, as determined by the jumper block, J_1.

A 2-3 jumper selects the internal divider, producing V_X=12V. A 2-1 jumper selects feedback resistors R_1 and R_2, producing $V_X = 1.5V(R_1+R_2)/R_2$. You should set V_X to 1 to 2V above the desired output voltage. The Q_1 linear regulator steps V_X down to an output level determined by R_3 and R_4: $V_{OUT}= 1.5V(R_3+R_4)/R_4$, where $V_X>V_{OUT}$. When $V_{IN}>V_X$, the switching regulator turns off, and the linear regulator alone controls V_{OUT}. C_6 reduces output ripple. The circuit accommodates a wide range of input and output voltages and supplies output currents as high as 500 mA (**Figure 2**). (DI #2218) EDN

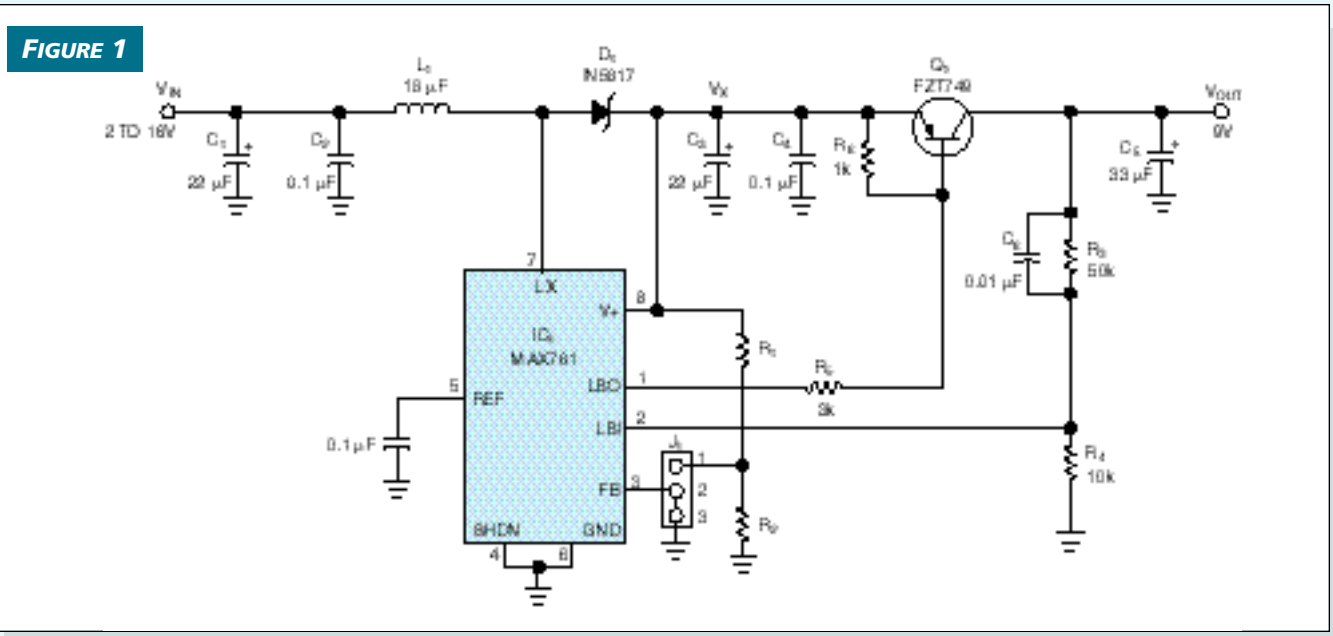
To Vote For This Design, Circle No. 418

FIGURE 2



Delivering more than 100 mA over its 2 to 16V useful input-voltage range, the regulator in Figure 1 provides 500 mA over an 8 to 13V range.

FIGURE 1



Toggling between switching and linear operation, this regulator operates with input voltages above and below the desired output voltage.

Battery booster delivers 75W

DONALD V COMISKEY, POWER TRENDS INC, WARRENVILLE, IL

The circuit in **Figure 1** defies no laws of physics; it just makes creative use of an isolated dc/dc converter. The application uses the isolated converter in a nonisolated configuration to boost a 48V battery voltage to 60V. The PT3102 is a 15W isolated dc/dc converter that normally uses a 36 to 75V (48V-nominal) input voltage to provide a floating, or isolated, 12V output capable of 1.25A. The trick is to connect the negative-output lead and positive-input lead of the converter, thereby effectively stacking the 12V output on top of the 48V input. (Be aware that the original isolation and safety properties of the converter no longer exist in this configuration.)

A load connected across the converter's positive-output and negative-input leads now sees a total output voltage of 60V. As **Figure 1** indicates, the load-current path is through the battery and the output section of the dc/dc converter. Assuming that the battery is capable of the same 1.25A as the converter, the total power available to the load is 60V31.25A, or 75W.

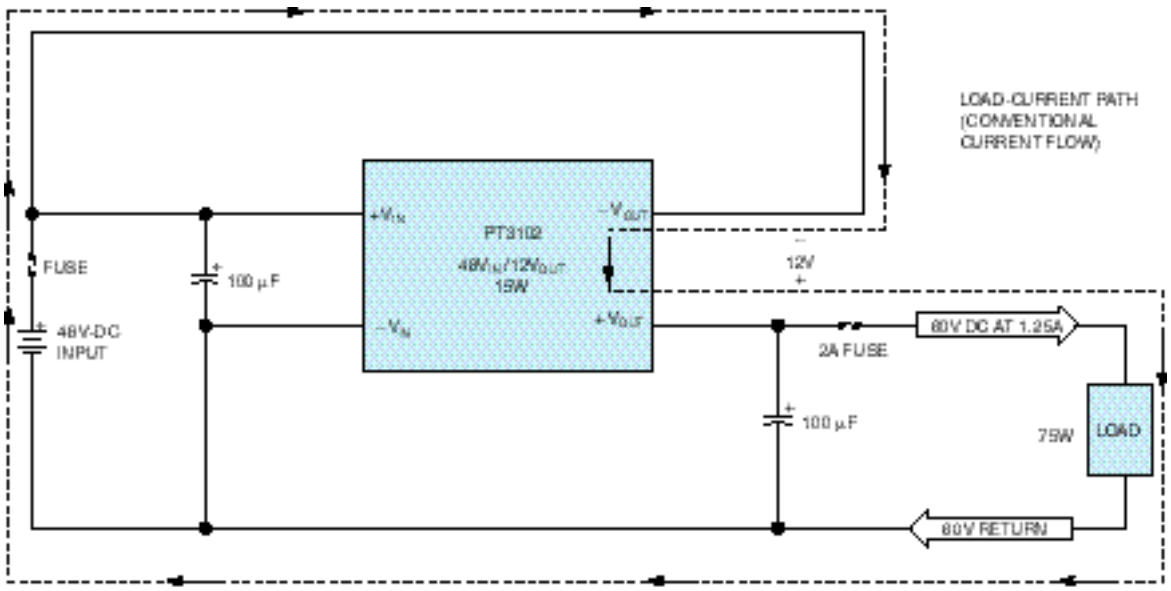
The dc/dc converter regulates its 12V output to within ±2%. The magnitude of the overall output voltage, however, depends on both the 12V output and the battery voltage. If the battery voltage droops to 40V, for example, the overall output voltage is 40+12=52V. If you replace the battery with a 48V power supply with ±5% regulation, the overall output voltage is 60V ±7%.

The dc/dc converter is 80% efficient, meaning it dissipates 3.75W while delivering its rated 15W (12V31.25A). When comparing this 3.75W of power dissipation with the load power of 75W, the overall efficiency is 95%.

The scheme isn't limited to 48V applications. Another application, for example, may need to boost a 24V battery to 36V. You can simply replace the PT3102 with the related PT3105 converter, which operates from 18 to 40V and provides an output voltage of 12V at 1.25A. This 12V output stacked on top of the 24V input provides 36V to the load and results in an efficiency of 3.75W/(36V31.25A)=92%.

In this stacking configuration, use of either converter's on/off control is not recommended, because activating this control succeeds only in disabling the 12V portion of the total output voltage. Even with a disabled converter, a path still exists through the output section of the converter, which comprises a forward-biased diode, causing the input voltage minus the diode drop to appear across the load. For similar reasons, you can't rely on the converter's built-in current

FIGURE 1



The negative-output lead and positive-input lead of the PT3102 dc/dc converter connect to effectively "stack" the converter's 12V output on top of the 48V input, providing 60V total across a 75W load.

limit for overload protection. Instead, use a 2A fuse in series with the output of the converter to protect against excessive load currents.

Because medium- and high-power boost converters are not commonly available off the shelf, custom or build-your-own designs are often necessary. This idea uses a common off-the-shelf converter, normally intended for step-down applications, to accomplish the boost function. Higher power applications, of course, require appropriately sized converters. If regulation of the overall output voltage is necessary, you can consider using an adjustable converter. (DI #2223) EDN

To Vote For This Design, Circle No. 419

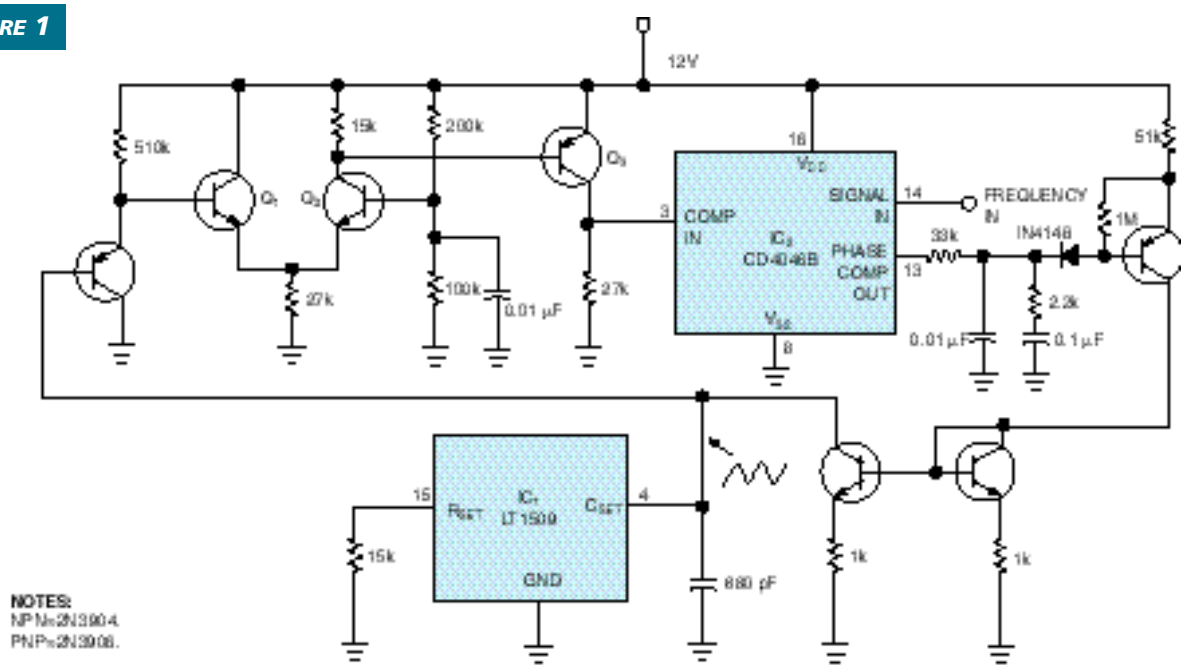
KURK MATHEWS, LINEAR TECHNOLOGY CORP, MILPITAS, CA

It is often desirable to synchronize a switching power supply to a system clock or to a second supply. If the power-supply controller has no synchronization pin, you can add a low-value resistor in series with the oscillator capacitor, thereby developing a synchronization pulse to sum into the oscillator ramp. Depending on the controller, this method may have drawbacks. Primarily, the wider the synchronization range, the higher the sync-pulse amplitude required (without exceeding the controller's internal reference voltage). Further problems arise if the controller IC uses the oscillator ramp for voltage-mode control, slope compensation, or timing. Under these circumstances, modifying the amplitude of the oscillator ramp may not be an option. Such is the case with IC_1 in **Figure 1**, an LT1509 PWM and power-factor controller, which depends on ramp amplitude for determining the phase relationship between the power-factor and PWM sections and for determining duty cycles.

The circuit in **Figure 1** uses a PLL to adjust the oscillator-ramp charge current, providing a wide synchronization range without changing oscillator voltage levels. Q_1 to Q_3 form a comparator that supplies a square wave (at the oscillator frequency) to IC_2. IC_2's phase comparator II output adjusts the oscillator-ramp charge current until the oscillator frequency equals the input frequency. The method provides synchronization and preserves the oscillator waveform and the IC's function. With the values shown, the frequency-lock range is 90 to 150 kHz. The circuit requires less than 1 mA from a 12V supply. (DI #2233) EDN

To Vote For This Design, Circle No. 420

FIGURE 1



A PLL provides frequency synchronization for a PWM and power-factor-controller IC.

Simple Spice model simulates laser diode

LUKASZ SLIWCZYNSKI, UNIVERSITY OF MINING AND METALLURGY, KRAKOW, POLAND

When you design drive circuitry for laser diodes, you must consider safety measures. Laser diodes are delicate devices, and excessive reverse voltage or forward current can easily destroy them. Usually, a laser-diode driver circuit comprises the laser diode and a monitor photodetector in a common package and a low-frequency feedback loop that stabilizes the diode's optical output power. Computer simulation of the driver before you turn it on can be helpful in thwarting laser-diode mortality.

Improperly designed driver circuitry can provoke laser-diode failure in two ways: large transients during the turn-on or -off phase and circuit instability. The first failure mechanism is unamenable to computer simulation, because it would require a reliable model of the power supply's switching behavior. However, even a laser diode with well-defined turn-on and -off transients may be in peril if its frequency characteristic shows high peaks or a tendency to oscillate. It's relatively easy to check oscillation with computer simulation using a Spice model. The model uses the simplified equations describing the relationship between the laser diode's current, the monitor's current, and the optical output power:

$$P_{LAS} = \begin{cases} 0 & \text{IF } I_{LAS} < I_{TH} \text{ AND } I_{MON} = S_{MON} P_{LAS} \\ \varepsilon (I_{LAS} - I_{TH}) & \text{IF } I_{LAS} \geq I_{TH} \end{cases},$$

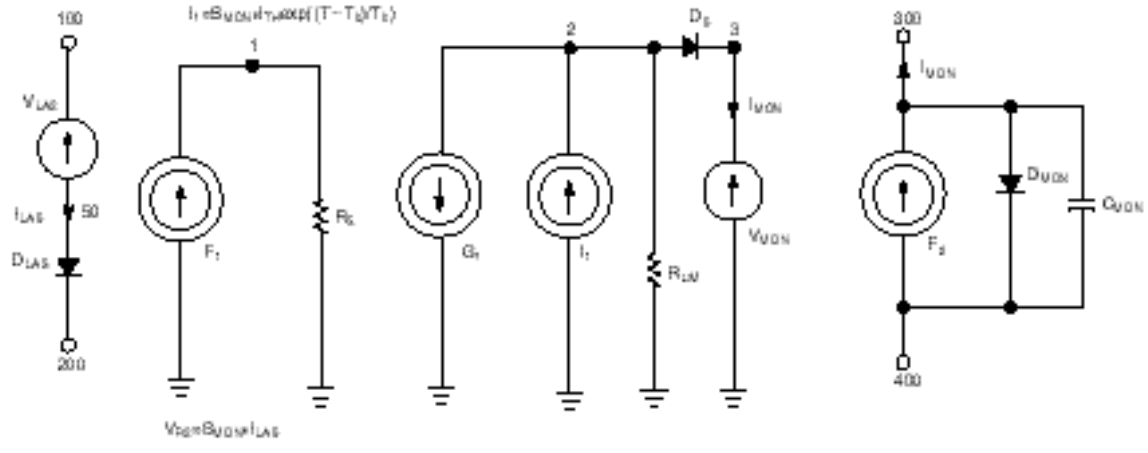
where
P_{LAS}=laser-diode output optical power,
I_{LAS}=forward laser-diode current,
I_{TH}=laser-diode threshold current,
I_{MON}=monitor-photodetector current (proportional to laser-diode power),

$$\varepsilon = \frac{dP_{LAS}}{dI_{LAS}}\Big|_{I_{LAS} \geq I_{TH}} = \text{LASER-DIODE SLOPE EFFICIENCY, AND}$$

$$S_{MON} = \frac{P_{LAS}}{I_{MON}} = \text{MONITOR-EFFICIENCY COEFFICIENT.}$$

Figure 1 shows the simple Spice model of the laser diode with its associated monitor photodetector. **Listing 1** gives the Spice netlist. The laser-diode pin assignments are 100 (anode) and 200 (cathode); the monitor assignments are 300 (anode) and 400 (cathode). Because the laser diode's junction operates forward-biased under normal conditions, the Spice model represents it as voltage source V_{LAS} in series with diode D_{LAS}. You should choose the value of V_{LAS} to match the emitted wavelength (approximately 1V for 820 nm and approximately 0.8V for 1300 nm). Current source I_1 models the laser diode's threshold-current-temperature dependence. R_{LIM} limits the voltage at Node 2 when $I_{LAS}<I_{TH}$. Diode D_S starts to conduct when the laser-diode current rises above the threshold

FIGURE 1



A simple Spice model can prevent destruction of laser diodes by providing a forewarning of dynamic instability.

value. Under normal operating conditions, the monitor photodetector is reverse-biased; for the purpose of ac analysis, the Spice routine models the photodetector as capacitance C_{MON}. This component is the only one affecting the dynamic behavior in this model, because the laser diode itself is much faster than any component in the driver circuit (1- to 3-GHz or higher cutoff frequency). You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2227. (DI #2227) EDN

LISTING 1—SPICE NETLIST FOR LASER/MONITOR DIODE PAIR

```
Vlas        100 50 1
Dlas        50 200 diode1

F1          0 1 Vlas 1
Rs          1 0 {Sm*eps}

I1          2 0 {Sm*eps*Ith*exp((T/25)-1)}
G1          0 2 1 0 1
Rlim        2 0 1Meg
Ds          2 3 diode
Vm          3 0 0

F2          300 400 Vm 1
Dmon        300 400 diode
Cmon        300 400 Cmon

.model      diode    d
.model      diode    d rs=5

.ENDS
```

References

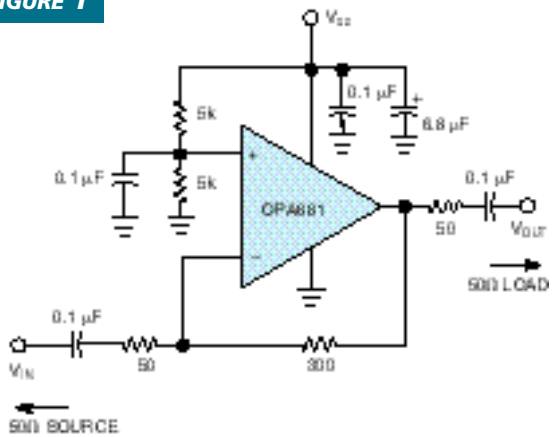
1. *Fiber Optics Handbook*, Hewlett-Packard, 1989.
2. Kressel, H, *Semiconductor Devices for Optical Communication*, Springer-Verlag, 1980.

To Vote For This Design, Circle No. 421

MICHAEL STEFFES, BURR-BROWN CORP, TUCSON, AZ

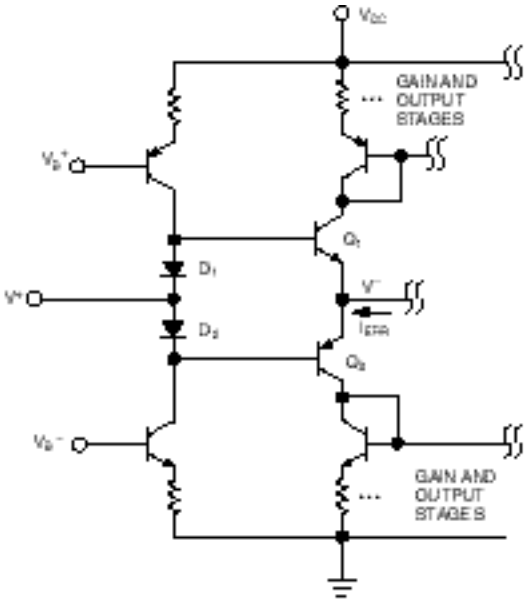
Most of the recent wideband, single-supply op amps use a voltage-feedback design. The error signal (differential input voltage) for a voltage-feedback op amp can operate over a range of common-mode input voltages, and its common-emitter output stage can provide output voltages near the supply rails. Current-feedback op amps are becoming available with wide output swings, but they still typically require at least 1.5V input head room for proper operation of the input stage's voltage buffer and inverting-node error-current

FIGURE 1



This current-feedback amplifier retains high bandwidth and a wide input-voltage swing, even with its input-stage transistors saturated.

FIGURE 2

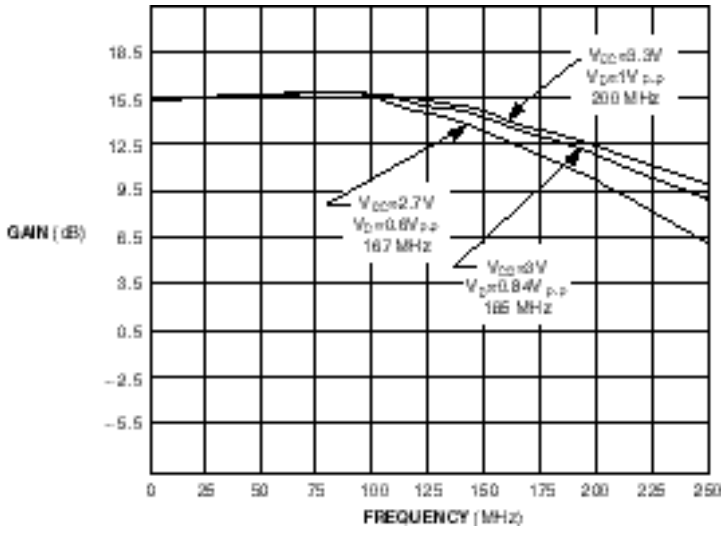


Saturation reduces the gain of Q_1 and Q_2, but the transistors still retain enough gain to provide adequate bandwidth in the amplifier.

sensing in the noninverting configuration. A current-feedback op amp offers the main advantage of wideband operation at higher gains. A subtler and previously overlooked advantage is the fact that you can operate most devices with a total supply voltage lower than the rated total input-stage head-room requirement. **Figure 1** shows a wideband, current-feedback op amp operating with a saturated input stage.

The OPA681 current-feedback op amp has a bandwidth greater than 200 MHz and a high-power output that can swing to within 1V of the supply rails. The input, however, requires 1.5V head room for proper operation. This requirement implies that, with a 3V power supply, the input stage is off while the output should still offer a 1V p-p swing. The circuit in **Figure 1** establishes a bias voltage on the noninverting input, set to the supply midpoint. The inverting signal path has an ac-coupled gain of 6. For test purposes, set the signal-input resistor to match the 50V source; the output drives a 50V matching resistor, through a dc-blocking capacitor, to a 50V load.

FIGURE 3



By ignoring the input-range specs of a current-feedback op amp, you can obtain extended wideband operation with low supply voltages.

In actual applications, you may not require this input match, and the output load may not need to be a doubly terminated line. However, the frequency response strongly depends on the selected value for the feedback resistor. Increasing it bandlimits the design; reducing it peaks the response. This circuit provides more than 150-MHz bandwidth, even for supplies lower than 3V, which would appear to violate the input-range spec. To understand this unique feature of a current-feedback topology, consider a simplified input-stage architecture in **Figure 2** .

If the noninverting input connects to $V_{CC}/2$ (**Figure 1**) and you steadily reduce the supply voltage, the first junctions to saturate are in Q_1 and Q_2. These two transistors provide the essence of the current-feedback operation. In normal operation, they provide voltage buffering for the input voltage (signal) at the noninverting input while acting as common-base stages to cascode the error-current signal into the inverting node through to the current-mirror gain stages. Consider what happens as these transistors saturate. The voltage-buffer aspect of the circuit still operates in a dc sense with a considerable drop-off in bandwidth if the signal were injected into the noninverting input. The error current still effectively cascodes through these transistors with a decreased a (the collector-emitter current gain in common-base configuration).

Under normal operation, a is nearly 1. As Q_1 and Q_2 saturate, a decreases, effectively reducing the dc open-loop gain for the forward signal path internal to the current-feedback amplifier. Because this open-loop gain is already high, an a even as low as 0.5 reduces the dc open-loop gain by only 6 dB with little effect on the open-loop frequency response. Q_1 and Q_2 can operate well into saturation, with little input on frequency response. This effect is a unique aspect of a current-feedback design, because similar operation for the differential input stage of a voltage-feedback op amp is impossible with the input-stage transistors saturated. You can use this feature to extend the range of low-voltage operation for current-feedback op amps that offer higher output swing than input range.

Figure 3 shows the results of tests with 3.3, 3, and 2.7V supplies. In each case, the output-voltage swing increased to just below the onset of gain compression. You can determine the required output-voltage head room for each supply voltage by subtracting the peak-to-peak output from V_{CC} and then dividing by two. You obtain slightly more output-voltage swing with light loads, such as an ADC input. With a 2.7V supply, the inverting-input transistors, Q_1 and Q_2 in **Figure 2**, are well-saturated, yet the amplifier still provides more than 150-MHz bandwidth with an inverting gain of 6V/V (15.5 dB from input to output), with a 0.6V p-p output swing. Small-signal operation maintains more than 100-MHz bandwidth with a supply as low as 2.2V. This bandwidth would have required an equivalent single-supply, voltage-feedback op amp with approximately 600-MHz gain-bandwidth product and a greater-than-300V/msec slew rate. (DI #2232) EDN

To Vote For This Design, Circle No. 422

CRAIG VARGA, LINEAR TECHNOLOGY CORP, MILPITAS, CA

To power a load that requires a high-compliance current source (for example, a plasma tube, such as an ion laser), a circuit must provide very accurate current control with a wide bandwidth and a voltage compliance of several hundred volts. You can use a high-speed linear-regulator controller to perform just such a function **Figure 1** . The bandwidth of this circuit measures more than 1 MHz. Because no part of the control circuit connects to the input supply, only the selected MOSFET and the ratings of the input supply's bypass capacitors limit the voltage compliance.

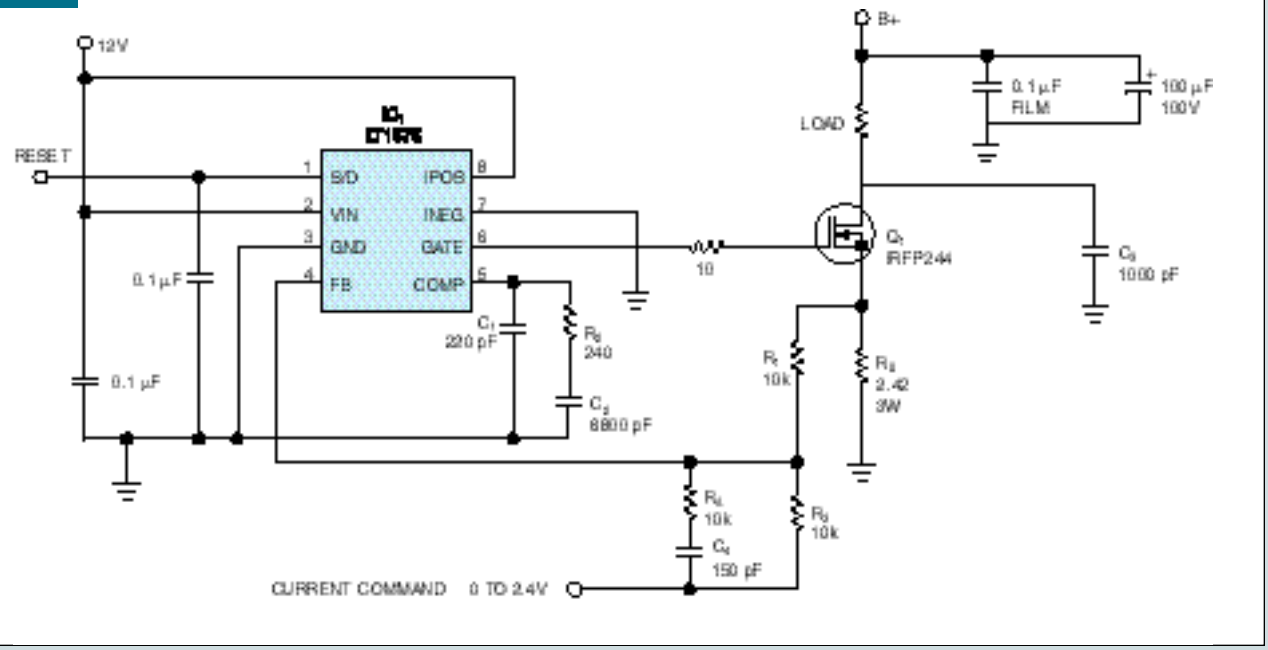
The typical application of the LT1575 linear regulator (IC_1) is to drive a MOSFET as an overdriven source follower, providing a high-speed and accurately regulated output voltage. The internal error amplifier and output buffer drive the large capacitive loads presented by MOSFET gates with an open-loop bandwidth of more than 15 MHz. You can implement a current source by replacing the output-voltage feedback divider with a current-sense resistor. The internal reference is nominally 1.21V with an accuracy of ±0.6%. Thus, you can program very accurate currents. IC_1 continuously adjusts the FET's gate voltage to maintain a constant load current, regardless of the voltage across Q_1. The input capacitors' voltage rating limits the circuit to 100V. The FET has a rating of 250V. Only the voltage ratings of the FET and input capacitor limit the circuit's compliance.

With a current-command voltage of 0V, the load current is approximately 1A. As the current command increases, the load current decreases. At approximately 2.42V, the load current drops to zero. By changing the values of R_1, R_3, and R_5, you can alter the scale factor. R_2, C_1, and C_2 provide loop compensation. R_4 and C_4 compensate for a slight drop in gain between approximately 80 and 200 kHz. These values are likely to depend on layout and may need adjusting in another implementation. When a high-compliance voltage is necessary, C_3 connects from the FET's drain to ground. The FET's drain-to-source capacitance decreases substantially at high stand-off voltages. C_3 swamps the FET capacitance, ensuring stability under all conditions.

Obviously, Q_1 needs a substantial heat sink if high voltage and even moderate currents are simultaneously present. If the power dissipation level becomes unmanageable, you can achieve a manageable level by making as many identical stages parallel as necessary to reduce each stage's current. (DI #2234) EDN

To Vote For This Design, Circle No. 423

FIGURE 1



A high-speed linear regulator (IC_1) can provide accurate current control with a wide bandwidth and a voltage compliance of as many as several hundred volts.

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____ Fax _____

E-mail _____

My e-mail address may be published Yes____ No____

Company _____

_____

Address _____

_____

Country _____ ZIP _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

The winning Design Idea for the December 4, 1997, issue is entitled "Current amp provides frequency modulation," submitted by Miguel Pereira of University of Vigo (Vigo, Spain).

EDITED BY BILL TRAVIS & ANNE WATSON SWAGER

Software PLL locks VCXO to reference

MARTIN OSSMANN, PHILIPS RESEARCH LABS, AACHEN, GERMANY

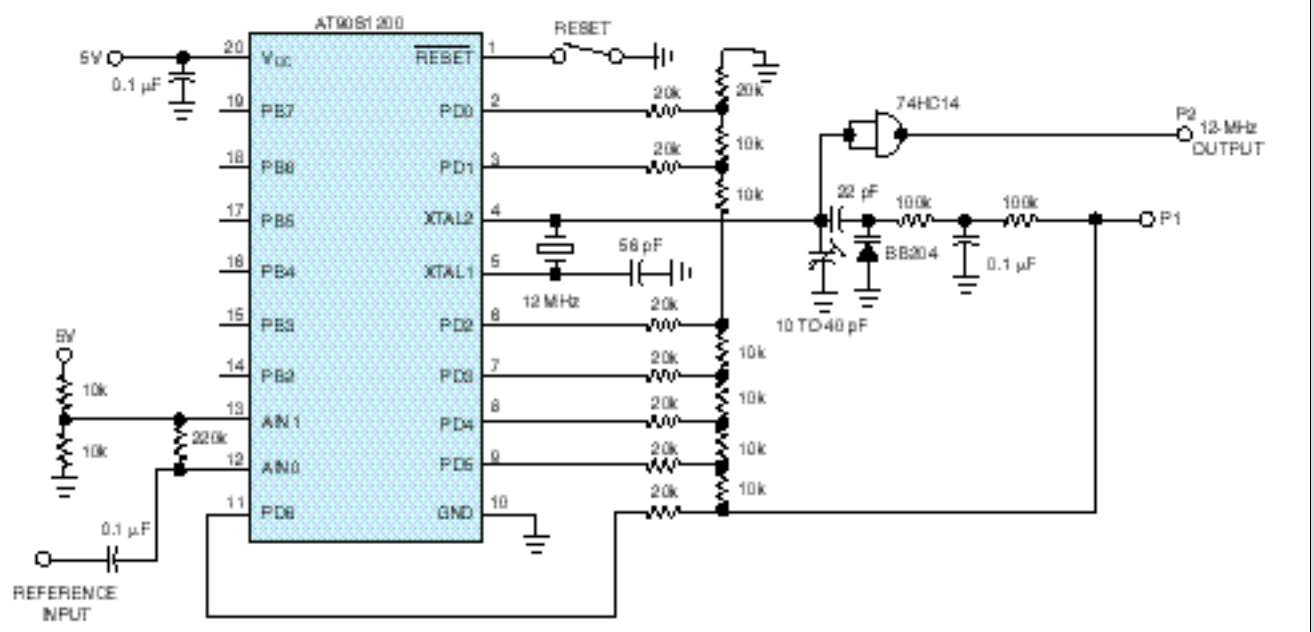
The circuit in **Figure 1** provides a precisely controlled clock signal. The method modifies the controller's quartz oscillator such that the control voltage at point P1 controls the frequency. A D/A converter using an R-2R ladder network connected to output port D generates the control voltage. A precise reference-frequency signal connects to the controller's analog-comparator input, AIN0. In this application, the reference frequency is 162 kHz (derived from a long-wave broadcast station). You can use any other 10- to 200-kHz frequency. The Atmel (www.atmel.com) AT90S1200 controller performs a PLL function to lock the clock to the reference. You can use the precise frequency available at point P2 to clock mCs or other digital circuitry. An 18-instruction program provides the PLL function (**Listing 1**). You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2229.

Figure 2 shows the operating principle. The variables dds0 to dds3 hold a 32-bit, direct-digital-synthesis (DDS)-type numerical oscillator. The routine XORs the

LISTING 1—AT90S1200 CODE FOR PLL FREQUENCY LOCK

```
;   EDN1.ASM:   lock the 12 MHz VCXO to a 162 kHz reference
      .device  at90s1200
      .include "1200def.inc"
.def    tmp     =r16
.def    cntr1   =r17        ; count-and dump time-counter
.def    phase   =r18        ; phase integrator
.def    dds0    =r26        ; LSB of 32 bit DDS register
.def    dds1    =r27
.def    dds2    =r28
.def    dds3    =r29        ; MSB of DDS register

RESET:
        ldi   tmp,$7F       ; port D set to OUTPUT-mode
        out   DDRD,tmp      ; port D is analog 7 bit pase output
SAMPLE:
        mov   tmp,dds3      ; perform DDS-MSB XOR with analog input
        sbic  ACSR,ACO
        com   tmp           ; conditional complement of MSB of DDS
        sbrc  tmp,7         ; conditional increment depending on XOR value
        inc   phase         ; compute phase
noSAMPLE:
        subi  dds0,$fe      ; 32 bit DDS function
        sbci  dds1,$d4      ; DDS rate is 12 MHz / 12 cycles = 1 MHz
        sbci  dds2,$78      ; increment is 162kHz/DDS_rate*2^32
        sbci  dds3,$29      ; is 2978D4FE (hexadecimal)
        subi  cntr1,1       ; integrate-and dump timer
        brne  SAMPLE        ; test for end of count
        clr   cntr1         ; reload timer
        lsr   phase         ; shift phase value to right
        out   PORTD,phase   ; output to 7-Bit DAC
        clr   phase         ; clear phase-value
        rjmp  noSAMPLE      ; and go on, one DDS-cycle is 12 clock periods
```

FIGURE 1

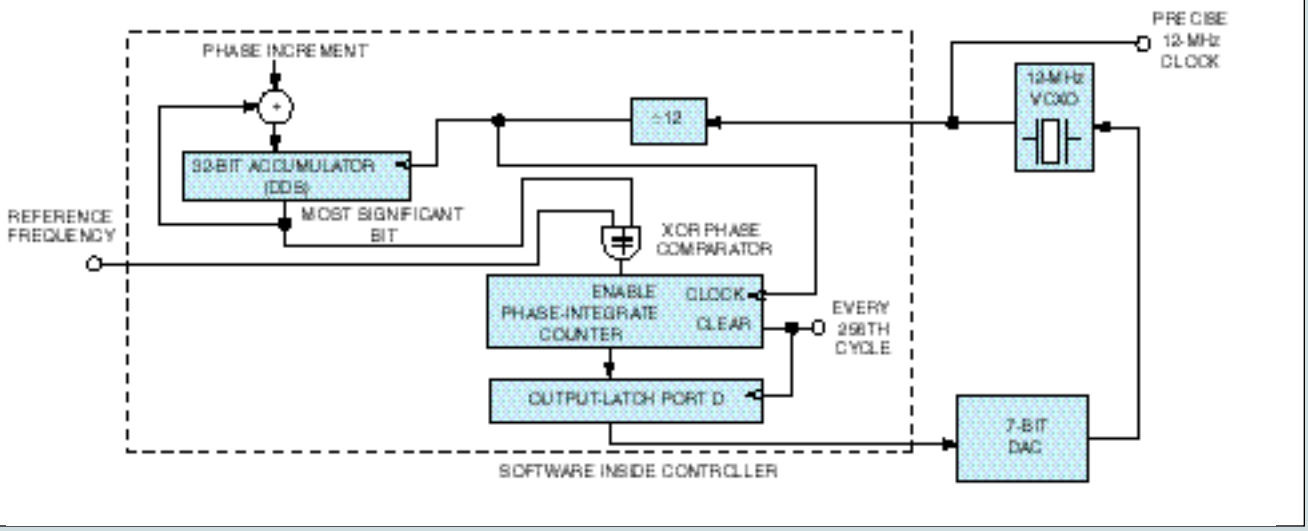


A microcontroller can provide a PLL function to obtain a precise reference-locked frequency.

most significant bit with the reference signal that enters the controller via the analog comparator. A count-and-dump function integrates the XOR output over 255 DDS cycles. Upon every 256th DDS cycle, the phase value routes to the DAC at Port D of the controller. Every loop lasts exactly 12 clock cycles. Thus, the DDS cycle frequency is one-twelfth of the controller's clock frequency. The program performs a phase comparison between the DDS and the reference. The DAC voltage controls the VCXO. By using other increments for the DDS, you can easily adapt the program for other clock or reference frequencies. You can also use the circuit for demodulating phase-modulated signals. (DI #2229) EDN

To Vote For This Design, Circle No. 301

FIGURE 2



The frequency-lock circuit in Figure 1 uses DDS techniques to provide a precise submultiple of the reference frequency.

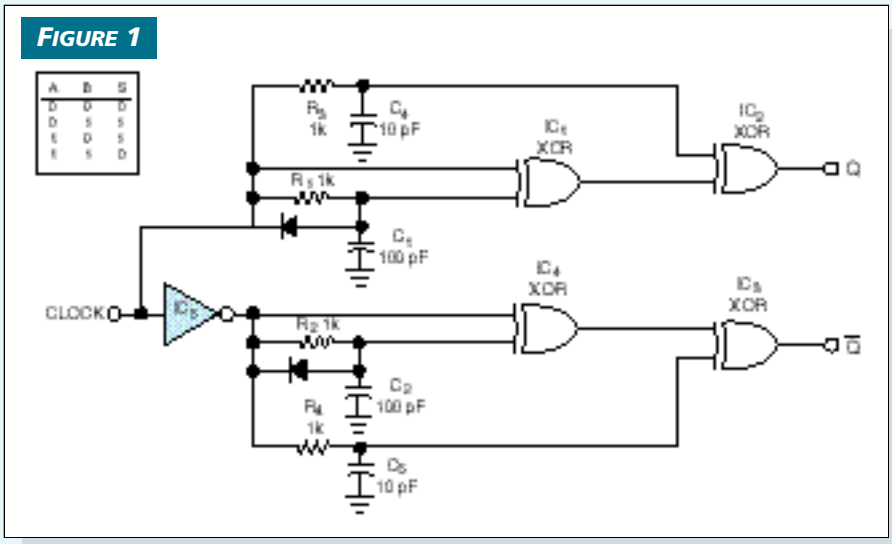
Spice introduces dead time in simulations

CHRISTOPHE BASSO, MOTOROLA SEMICONDUCTOR, TOULOUSE, FRANCE

Bridge or half-bridge designs using MOSFETs or insulated-gate bipolar transistors need some dead time between commutations to avoid any cross-conduction current spikes. This statement is also valid for switch-mode power supplies that use synchronous rectification. In creating simulations, it is sometimes difficult to write the stimuli so as to define a dead time between commutations. Classic PULSE or PWL commands are impractical, especially when either frequency or pulse width changes during the simulation run. **Figure 1** shows a approach to simulating dead time that

An inverter, a few XOR gates, and some passive components generate a dead-time interval for switch commutation.

FIGURE 1

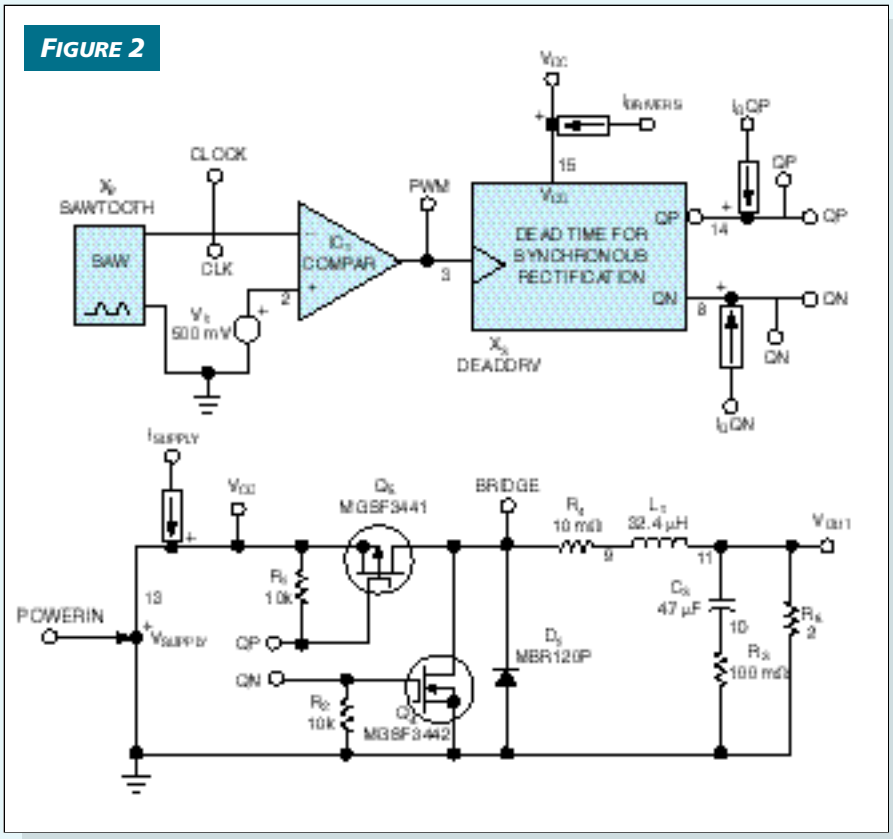


you can build around a few logic XOR gates. The principle uses the truth table of an XOR or XNOR gate: that the output is high or low only when both inputs have different logic states.

The logic states come from the RC networks R_1-C_1 and R_4-C_5. The output of the IC_1 and IC_4 gates is thus a short pulse whose width depends on the RC time constants of the input network. This pulse blanks the signal delivered to the output and thus generates the required dead time. You can easily model the logic functions using Intusoft's (www.intusoft.com) IsSpice4 Analog Behavioral Modeling features (**Listing 1**). You need to feed the subcircuit with the dead-time value as well as the output high and low levels. The input clock is TTL/CMOS-compatible. By changing the B5 line to V=V(26,20)<100MV ? {VHIGH} : {VLOW}, the generator becomes suitable for driving a synchronous rectifier (**Figure 2**). **Figure 3** clearly shows the absence of overlap between commutations. You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DISIG, #2228. (DI #2228) EDN

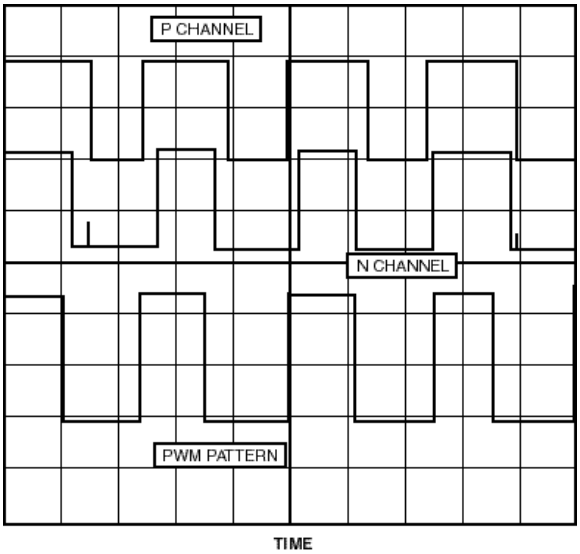
To Vote For This Design, Circle No. 302

FIGURE 2



The circuit in Figure 1 generates the dead time to prevent cross-conduction in this synchronous-rectifier circuit.

FIGURE 3



Current waveforms for the MOSFETs in Figure 2 show no simultaneous conduction.

LISTING 1—ISSPICE4 ANALOG BEHAVIORAL MODELING FEATURES

```
.SUBCKT DEADTIME 1 50 51 {DT=500N VHIGH=10V VLOW=100M RS=10}
* Clock_In Q Qbar
* Developed by Christophe BASSO (FRANCE)
RIN 1 0 1MEG
B6 17 0 V=V(1)>2V ? 10 : 0
R3 17 18 1k
C3 18 0 {DT/(1000*4.14)}
B4 21 0 V=V(25,19)<100MV ? {VLOW} : {VHIGH}
RCQ 21 60 100
CCQ 60 0 10P
BQ 61 0 V=V(60)
RSQ 61 50 {RS}
R4 22 23 1k
C4 23 0 {DT/(1000*4.14)}
B5 24 0 V=V(26,20)<100MV ? {VLOW} : {VHIGH}
RCQB 24 70 100
CCQB 70 0 10P
BQB 71 0 V=V(70)
RSQB 71 51 {RS}
R5 17 25 1k
C5 25 0 {DT/(1000*41.4)}
R6 22 26 1k
C6 26 0 {DT/(1000*41.4)}
D3 23 22 DISCH
D4 18 17 DISCH
B1 22 0 V=V(1)>2V ? 0 : 10
B2 19 0 V=V(17,18)<100MV ? 0 : 10
B3 20 0 V=V(22,23)<100MV ? 0 : 10
.MODEL DISCH D BV=100V CJO=4PF IS=7E-09 M=.45 N=2 RS=.8
+ TT=6E-09 VJ=.6V
.ENDS
```

Circuit translates TTY current loop to RS-232C

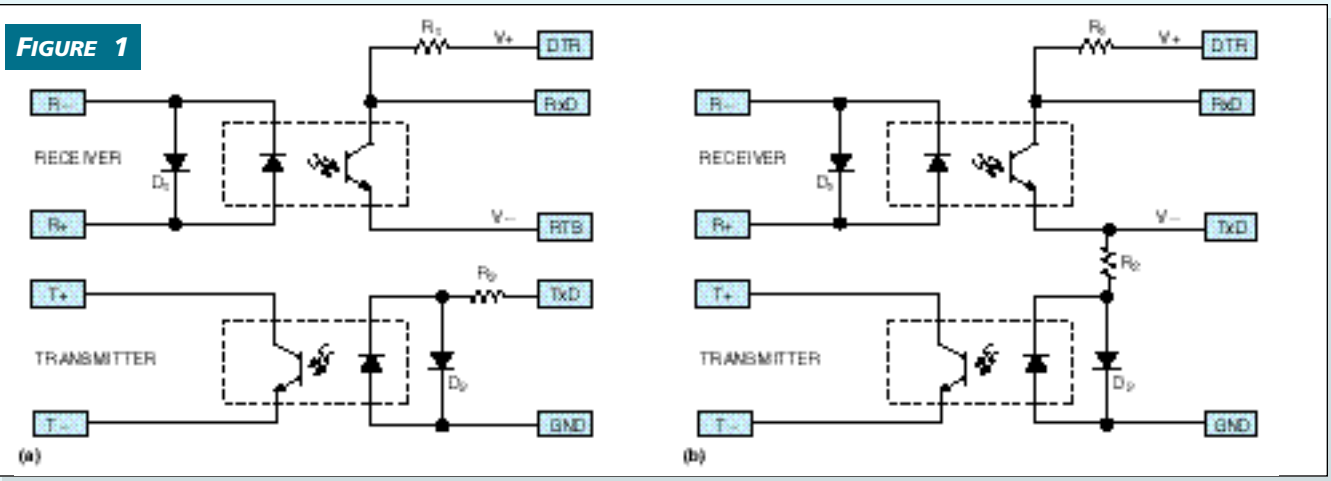
JERZY CHRZASZCZ, WARSAW UNIVERSITY, POLAND

The circuit in **Figure 1a** provides signal translation between a passive current-loop (TTY) interface and a duplex RS-232C port. The current flowing in the receiver loop causes the transistor to pull down RxD; when the transistor turns off, R_1 pulls up RxD. In like manner, the current in the transmitter loop switches on for a negative TxD voltage and off for a positive voltage. The supply power comes from the interface control lines, so you must properly preset these lines. Unfortunately, terminal programs do not usually support direct control of handshake signals. In other words, you must write your own service routine for the serial port to obtain a negative supply.

Worse, some RS-232C-like ports feature just one handshake line, rendering the circuit in **Figure 1a** unusable. In such cases, you could use the interface in **Figure 1b** . Because the transmitter output acts as the negative supply rail, the circuit can receive data only when TxD remains inactive. This limitation obviously precludes full-duplex transmission. Note also that transmitted data directly echoes at the RxD input. However, if half-duplex operation is satisfactory and you can tolerate local echo, this circuit may be the one of choice. In both circuits, external diodes (for example, 1N4148) protect the LEDs against reverse voltages. The values of R_1 and R_2 depend on the optocoupler type and loop current. You can use CNY75B optocouplers with 5.1 and 220V for R_1 and R_2, respectively. (DI #2230) EDN

To Vote For This Design, Circle No. 303

FIGURE 1



An optocoupler and two resistors configure a TTY-to-RS-232C translator. Use the circuit in (a) when a negative supply is available; otherwise, use the circuit in (b).

80186 timer pins provide general I/O

SK SHENOY, NPOL, KOCHI, INDIA

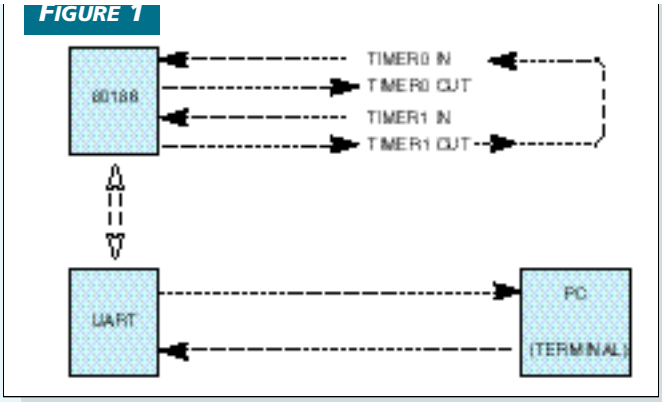
Intel's (www.intel.com) 80186 is a highly integrated mP common in embedded applications. It combines 15 to 20 common iAPX86 system components, such as a DMA controller, an interrupt controller, timers, a clock generator, a bus interface, and chip-select logic on one chip. Unfortunately, unlike many mCs, it provides no general-purpose I/O pins. However, if you require only a couple of input or latched-output lines, you can use the built-in timer I/O pins as general-purpose input or output lines by using this programming method (**Listing 1**).

The 80186 has two timers (0 and 1), each with two external pins: one for input, one for output. However, the structure of the timers allows you to use any pair of pins, either for input or output, one at a time, but not simultaneously. In other words, you cannot use Timer1_Out pin as a latched output and Timer1_In pin as an input at the same time. This limitation exists because, when you use Timer_In for input, the state of Timer_Out may change. Thus, you can simultaneously obtain two latched outputs, two input lines, or one input and one output line. Finally, a lag of a few microseconds occurs for an output to respond, because the output is the result not of a single "Out" instruction but of a sequence of instructions.

Moreover, you can see from the code in **Listing 1** that in some cases one or two timer ticks must elapse before the state changes. The same situation exists in sampling an input level.

```
/*
For compiling and linking; the following make batch file is used.
libs file contains names of library files which are to be linked.
Note that all addresses are system specific.

ic86 %1.c
link86 %1.obj,&<libs
loc86  %1.lnk ss(stack(+4096)) ad(cs(code(01030h),data(05040h))) ST(main)
*/

#pragma ram
#pragma fp
#pragma la
#pragma mod186
#pragma extend

typedef unsigned char byte;
typedef unsigned int word;

/* IC86 Specific header file */
#include  <icc\i86.h>

void delay(word count)
{
     word i;
     for (i=0;i <count;i++);
}

/* System specific addresses and utility routines for serial IO */
#include        <include\addr.h>
#include        <include\SccAsync.c>

/* Starts and stops the counter with Max Reg A in use
   so that timer1_out is High */
void    MakeHi()
{
        outword(0xff5e,0x4004); /* Stop timer 1 */
        outword(0xff58,0x0);    /* Count  = 0 */
        outword(0xff5a,0xffff); /* Large Max Count A */
        outword(0xff5e,0xc000); /* Start Counter, with int clk, A only, No cont 0xc004 */
        outword(0xff5e,0x4004); /* Stop Timer */
}

/* Starts the timer with Max Reg A and B alternating, Max Reg A is
   set for only 1 count and B for a large count. The timer is stopped as
   soon as Max Reg B is in use so that timer1_out is Lo */
void    MakeLo()
{
        outword(0xff5e,0x4003); /* Stop Timer1 */
        if ((inword(0xff5e) & 0x1000) != 0x1000)
        /* If output is now high; ie. A is in use, Make output Lo */
        /* Else output already Low; do nothing */
        {
                outword(0xff58,0x0); /* Count  = 0 */
                outword(0xff5a,0x1); /* Max Count A = 1 */
                outword(0xff5c,0xffff);/* Max Count B = Large value */

                 /* Start Counter, with int clk, Max Reg A&B, not continuous mode */
                 outword(0xff5e,0xc002);
                 /* Wait till B in use ie. Output goes Low */
                 while ((inword(0xff5e) & 0x1000) != 0x1000);
                 outword(0xff5e,0x4002); /* Stop Timer */
        }
}

/* Timer0 is kept running with RTG bit = 0 so that the timer is turned
   on or off depending on whether the Timer0_in is Hi or Lo. This routine
   tests the state of Timer0_in pin by checking if the timer is incrementing */
byte    ReadInput(void)
{
        outword(0xff50,0); /* reset count register */
        delay(3);
        if (inword(0xff50) > 0) /* if counting */
                return(1); /* Timer0_in is High */
        else return(0);
}

main()
{
        byte string[300];
        byte chr;

        init_stack();   /* Assembly routine for initialising stack pointer SP and base SS .*/
        outword(0xff56,0xc001); /* Start timer0 in continuous mode,RTG=0 */
        while (1)
        {
                PrintString("Enter Level to be Output(1/0):");
                chr = GetChar();
                if(chr == '1')
                {
                        PrintChar('1');
                        MakeHi(); /* Output Level 1 */
                }
                else
                {
                        PrintChar('0');
                        MakeLo(); /* Output Level 0 */
                }
                PrintChar(' ');
                PrintString("Input Level is  = ");
                if(ReadInput())
                        PrintChar('1');
                else PrintChar('0');
                crlf();
        }
}
```

FIGURE 1



You can use the timer pins in an 80186 mP to obtain a gener-

TABLE 2—DEFAULT 80186 TIMER REGISTER ADDRESSES

Register name	Timer0	Timer1	Timer2
Mode/control word	FF56H	FF5EH	FF66H
Max count B	FF54H	FF5CH	Not present
Max count A	FF52H	FF5AH	FF62H
Count register	FF50H	FF58H	FF60H

However, in all but the most demanding applications, this delay of a few microseconds should be unobjectionable. For such applications, this technique can save on the additional hardware you need to provide an external port. The demo program uses the setup in **Figure 1** . To output a bit, the routine makes the Timer1_Out pin assume either a 1 or 0 state by activating MaxCount A or B, respectively, and stopping the timer in that state. The state of the Timer_Out pin reflects which Max register the timer uses (register 1 for Max Reg A).

To read an input fed to the Timer0_In pin, the routine keeps Timer0 running. Timer0 is configured so that it counts only if the Timer0_In pin is high. Thus, if Timer0 increments during two consecutive reads separated by a small delay greater than a timer clock period, it means the input level is high. **Listing 1** is a demo C program for obtaining output via the Timer1_Out pin and inputs via the Timer0_In pin. The program sends logic 1 or 0 to the Timer1_Out pin, based on the key (1 or 0) the user presses. If Timer1_Out connects to Timer0_In, as the dashed line in **Figure 1** shows, the program reads and displays the input's logic level through the Timer0_In pin. In the demo system, an RS-232C port on the 80186 hardware, connected to a terminal, serves for user keyboard input.

The demo C program is written and compiled using an Intel IC86 compiler on an 8-MHz 80186 system. However, the same technique is applicable to other mPs/mCs having similar timer capabilities. **Table 1** gives the timer control-word bits. **Table 2** provides the default 80186 timer register addresses.

You can download **Listing 1** from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2231. (DI #2231) EDN

To Vote For This Design, Circle No. 304

TABLE 1—TIMER CONTROL-WORD BITS

Bit 0	Set to 1 for continuous-mode running; 0 for one-shot mode
Bit 1	1 for time to alternate between maximum-count register A and B; 0 for A only
Bit 2	1 to select external clock for the timer; 0 for internal, that is, CPU clock/4
Bit 3	If 1, Timer 2 output is used as clock, else internal clock (CPU clock/4) is used
Bit 4	If 0, the input level gates the timer on or off (timer will count for a high)
Bit 5	This is a read-only bit set when the timer reaches its maximum value
Bit 11	This bit has to be 0
Bit 12	This read-only bit indicates which maximum-count register is in use (0 indicates A)
Bit 13	If set, interrupts are generated on every terminal count
Bit 14	If 0, Bit 15 (enable/disable) is ignored, else Bit 15 will take effect
Bit 15	If set, the timer is enabled; 0 stops the timer

Notes: All bits except bits 5 and 12 are read/write.
Bits 5 and 12 are read-only.
Bits not shown don't care.

Linearizing an RTD is easy with Spice

RICHARD FAEHNRICH, CONSULTANT, ARLINGTON HEIGHTS, IL

A popular way to design an accurate temperature probe using an inherently nonlinear device—a resistance temperature detector (RTD)—uses positive multiplicative feedback. However, the transfer function of this method is also nonlinear, making an analytical approach difficult. A simpler method uses Spice to optimize the RTD's response by exciting the sensor with a particular waveform and performing simple signal processing on the circuit's output. You can also measure the output's nonlinearity while quickly trying a range of feedback parameters.

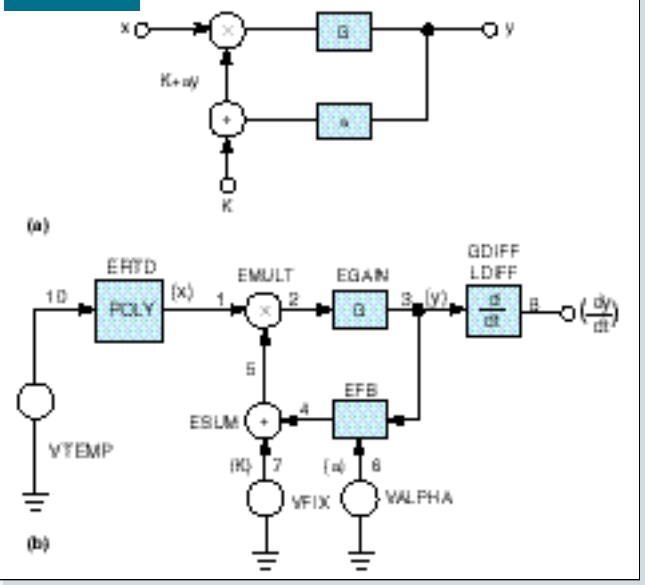
The block diagrams of the multiplicative-feedback circuit (**Figure 1a** and **Reference 1**) show how the scheme multiplies the input by a fraction of the output and a fixed term. The transfer function is

$$y = \frac{x \cdot K \cdot G}{1 - (x \cdot \alpha \cdot G)}.$$

Notice that when the feedback factor, a, is zero, the gain is simply equal to x•K•G. To understand the method, assume initially that x is linear. Introducing a small positive or negative a generates an output that is concave upward or downward, respectively. Now, if the input x exhibits an upward or downward curve, then the proper choice of a bends the curve in the opposite direction to improve the overall linearity.

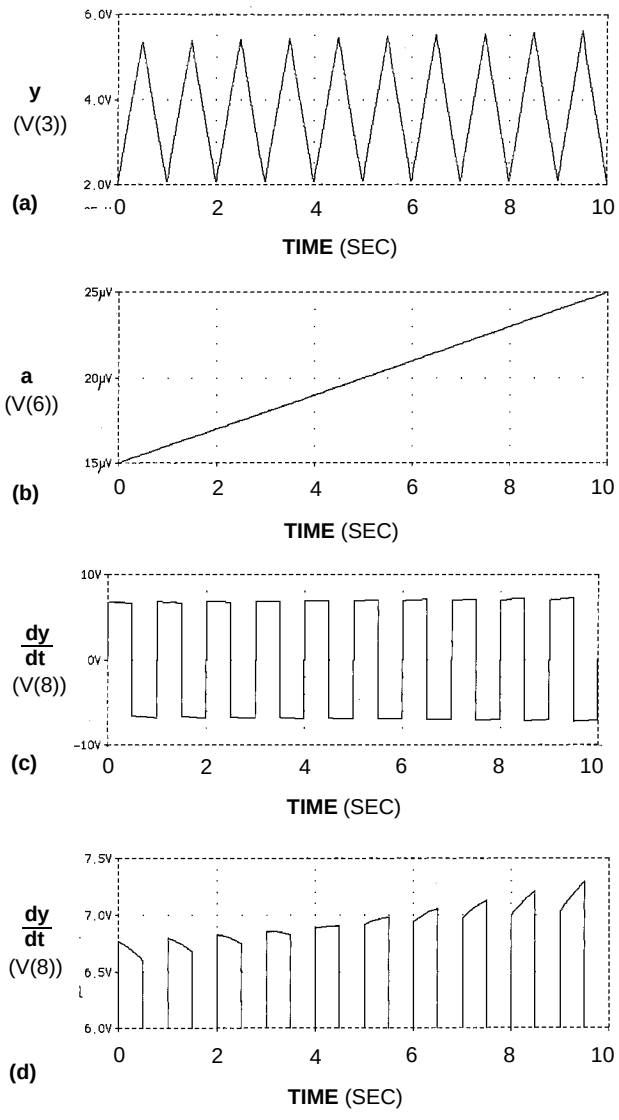
One way to measure the nonlinearity of the output is to fit a straight line to the curve and measure the maximum deviation of the output curve from this line. Another way is to perform a quadratic fit and monitor the second-order coefficient as a measure of straightness. Although both of these methods work, extracting the data from Spice output files

FIGURE 1



Multiplicative feedback (a) is one way to linearize an RTD. You can drive a corresponding Spice circuit (b) with a triangle waveform to measure the circuit's nonlinearity and determine the value of a, for which the derivative of the output is flattest.

FIGURE 2

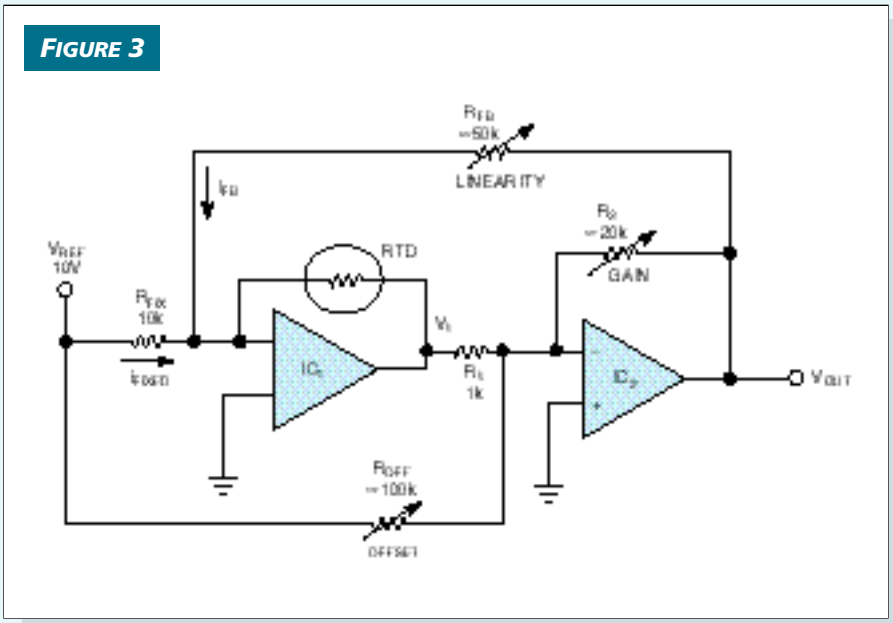


Spice makes it easy to view the output (a), the value of a (b), and the derivative of the output for two horizontal scales (c and d). The optimum value of a is the point at which the derivative is flattest, which occurs approximately 3 to 4 sec into the simulation, or when a '18 mV.

and curve-fitting over a range of values for a are difficult and time-consuming.

Alternatively, you can obtain some measure of nonlinearity by sweeping the temperature with a linear ramp function and then taking a derivative of the output response. A linear output should produce a constant derivative, and a concave upward or downward response produces an increasing or decreasing derivative, respectively. Therefore, the goal of this optimizing method is to find the value of a that results in the flattest derivative. However, because the input-ramp function increases without bound, this method instead uses a triangle wave to sweep the temperature. Now, by slowly sweeping a over many cycles of the triangle wave, a single simulation can measure the circuit's performance with varying parameters.

FIGURE 3



A practical implementation of the RTD linearizing circuit includes linearity, gain, and offset adjustments.

In the Spice circuit, VTEMP drives the input temperature linearly from 0 to 4008C with a 1-Hz triangle waveform (**Figure 1b**). (You can download the corresponding Spice netlist from *EDN*'s Web site, www.ednmag. com. At the registered-user area, go into the "Software Center" to download the file from DI-SIG, #2224.) ERTD simulates the RTD's response versus temperature using a second-order polynomial. To counteract the RTD's concave-downward behavior, the circuit introduces a positive feedback factor, a, with a concave-upward characteristic. VALPHA sweeps a linearly for 10 sec. Therefore, the simulation determines the performance of the circuit for the a range over 10 cycles of the triangle wave. The sum of the positive-feedback term, EFB, and a fixed term, VFIX, controls the gain of the multiplying block, EMULT. EGAIN provides a gain of G. Current source GDIFF and inductor LDIFF form the differentiator.

The linearized output and the a parameter appear in **Figure 2a** and **b**, respectively. **Figure 2c** shows the output derivative, and **Figure 2d** shows only the tops of the derivative's positive cycles. Note how the derivative in **Figure 2d** flattens, surrounded by a decreasing and increasing derivative to the left and right of the optimal behavior, respectively. You can start with a broad range for a and then quickly narrow the search after several simulations. Note that sweeping alpha during the triangle-wave cycles has some influence on the output response and, consequently, its derivative. However, this effect lessens as you narrow the range of a.

You can implement the positive-feedback factor as a potentiometer with an adjustment range that brackets the optimum value as determined by your simulations. Note that the overall gain and offset of the output versus temperature vary as a varies. Therefore, the final circuit should include a gain and offset adjustment in addition to the linearity adjustment.

In one suggested circuit implementation, you repeat the three adjustments until the circuit achieves the correct output voltages for the RTD resistances corresponding to the low, midpoint, and high calibration temperatures (**Figure 3** and **Reference 2**). This circuit realizes the fixed gain and feedback terms as currents. Op amp IC_1 implements the multiplier by first summing the fixed and feedback currents and then multiplying the result by the RTD's resistance. IC_2 provides additional gain, G, producing an output voltage given by

$$V_{OUT} = R_{RTD} \langle (i_{FIXED} + i_{FB}) \langle G,$$

where $G=R_2/R_1$. The feedback-factor calculation is

$$\alpha = \frac{i_{FB}}{V_{OUT}} = \frac{1}{R_{FB}}.$$

A feedback factor of a =0.00002 (R_{FB}=50 kV) in **Figure 3** reduces the linearity error from ±3.08C to less than ±0.18C over a 4008C range. (DI #2224) EDN

References

1. Sheingold, D, "Transducer Interfacing Handbook," Analog Devices, Norwood, MA, 1980, pg 99.

2. "The Linear Applications Handbook," 1990, Linear Technology, Application Note 6, pg 2.

To Vote For This Design, Circle No. 305

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02458

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____

E-mail _____ Fax _____

Company _____

_____

Address _____

_____

Country _____ ZIP _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

CIRCLE NO. 10

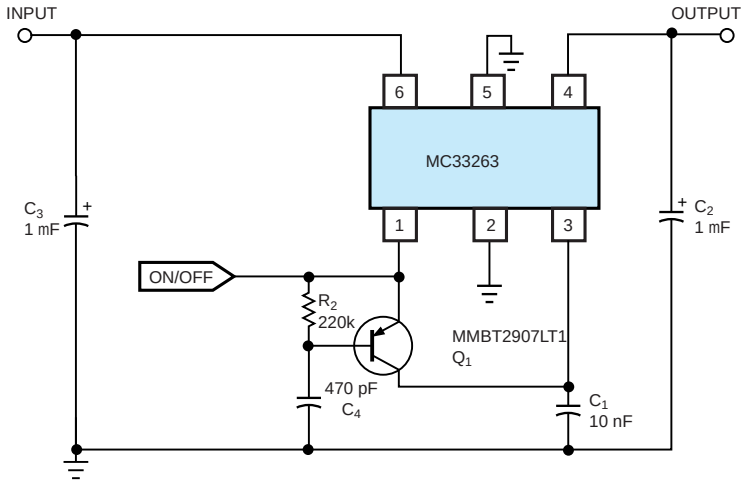
EDITED BY BILL TRAVIS & ANNE WATSON SWAGER

Transistor quickly wakes sleeping LDO

CHRISTOPHE BASSO, MOTOROLA SEMICONDUCTOR, TOULOUSE, FRANCE

Portable systems, such as telephone handsets, make extensive use of low-dropout (LDO) regulators. These components provide noise-sensitive parts with a stable power-supply line. When a telephone enters standby mode, most of the circuits go to sleep by disabling the LDO's outputs. Operating current thus drops to a minimal level. When a user starts to dial a number, the LDO receives an enable signal and immediately delivers the nominal operating voltage. Unfortunately, most low-noise LDOs use a bypass capacitor that briefly loads the internal reference voltage upon wake-up. In fact, the output exhibits a latency period before reaching its steady-state level. With a 10-nF bypass capacitor, this period typically lasts 1 msec and correspondingly degrades the overall response time. The ultra-low-noise MC33263 from Motorola (www.motorola.com) also uses a 10-nF bypass capacitor. However, the EZCap architecture of the IC allows the use of an inexpensive decoupling capacitor (ESR from 10 mV to 3V) and allows the designer to speed the wake-up time (**Figure 1**).

FIGURE 1



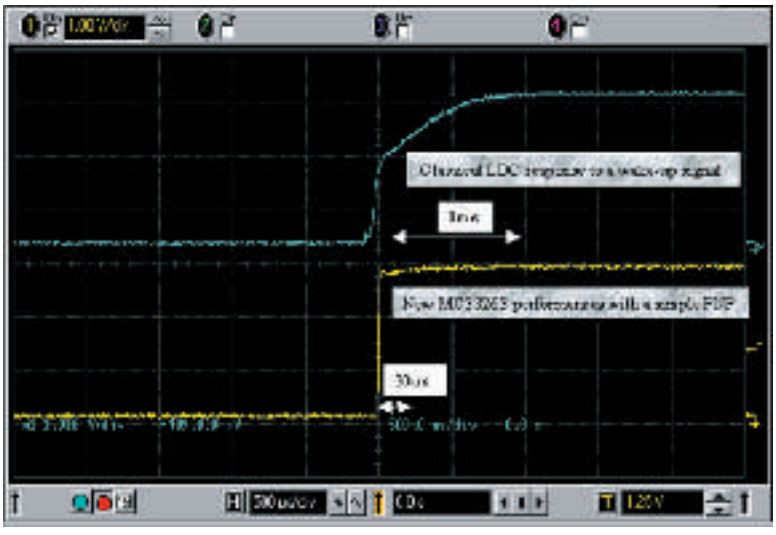
An inexpensive transistor and two passive components drastically improve the wake-up-response time of a linear regulator

The base of a low-cost pnp transistor connects to an RC network. At power-up, C_4 discharges. When the control logic sends its high-going wake-up signal, the transistor's base is momentarily tied to ground. The transistor turns on and immediately charges bypass capacitor C_1 toward its nominal operating voltage. After a few microseconds, the pnp turns off and becomes transparent to the regulator. This circuit dramatically improves the response time of the regulator from 1 msec to 30 msec (**Figure 2**). You need only adjust the RC time constant to avoid any bypass-capacitor overload during the wake-up transient. Such an overload would generate an unacceptable output overshoot. Because the transistor connects to the bypass pin, it does not degrade the noise performance of the LDO. (DI #2241)

e

To Vote For This Design, Circle No. 355

FIGURE 2



The simple circuit in Figure 1 improves the wake-up-response time of the LDO from 1 msec (upper trace) to 30 msec (lower trace).

Circuits provide 4- to 20-mA PWM control

TOM GAY, DARMSTADT, GERMANY

The circuits in **Figures 1** and **3** are useful when you use 4- to 20-mA current-loop signals to control a PWM signal. In both circuits, the minimum pulse width (corresponding to a 4-mA loop current) and the maximum pulse width (corresponding to a 20-mA loop current) are independently adjustable in the dedicated application with the use of one reference voltage. Furthermore, the circuits shut down the PWM output signal in case of a loop break. Both circuits are low-cost; you can use any op amp that provides an adequate slew rate to handle the desired PWM frequency.

The PWM circuit in **Figure 1** uses free-running oscillation. Amplifier IC_1 is a noninverting integrator that forces a constant current, I_C, into C_1, thus providing a constant linear slope on its output, V_{PWM}. When the divided fraction of V_{PWM} reaches the level of V_C on the negative input of comparator IC_3, the comparator's output switches high. Hence, FET Q_2 short-circuits divider resistor R_5, and the undivided level of V_{PWM} undergoes comparison with V_C. R_7 extends the turn-on time of Q_1 with respect to Q_2, such that FET Q_1 discharges C_1 with minimum delay. The output of amplifier IC_1, V_{PWM}, then rapidly slews down to the level of V_{R4}.

Consequently, the output of comparator IC_3 switches back to logic low, and IC_1 restarts charging C_1. To ensure that IC_3 returns to a logic-low level, you must generally set V_C a few millivolts higher than V_{R4}. The divider network, R_2, R_3, and R_4, guarantee this setting. You can use the output of comparator IC_3 as a trigger signal for synchronizing other circuits. Resistor R_{IN} terminates the loop current, and comparator IC_2 provides the PWM circuit's output signal, PWM_{OUT}, by comparing the current-loop signal V_{IN} with V_{PWM}. The circuit in **Figure 1** has PWM_{OUT} set to 0% duty cycle at 4-mA loop current and 80% at 20-mA loop current (**Figure 2**).

The circuit in **Figure 3** is a simplified, gated version of the circuit in **Figure 1**, with a synchronization input. It works the same, with one difference: The output of amplifier IC_1, V_{PWM}, rises to and remains at the positive supply rail for IC_1. As before, FET Q_1 short-circuits C_1 with each rising edge on the Sync input, causing IC_1 to slew V_{PWM} down to the level of V_{R4}. The network comprising R_2, R_5, and C_2 transforms a low-to-high transition on the Sync input into a narrow gate pulse, which turns on Q_1 for a short time. If you connect the Sync output of the

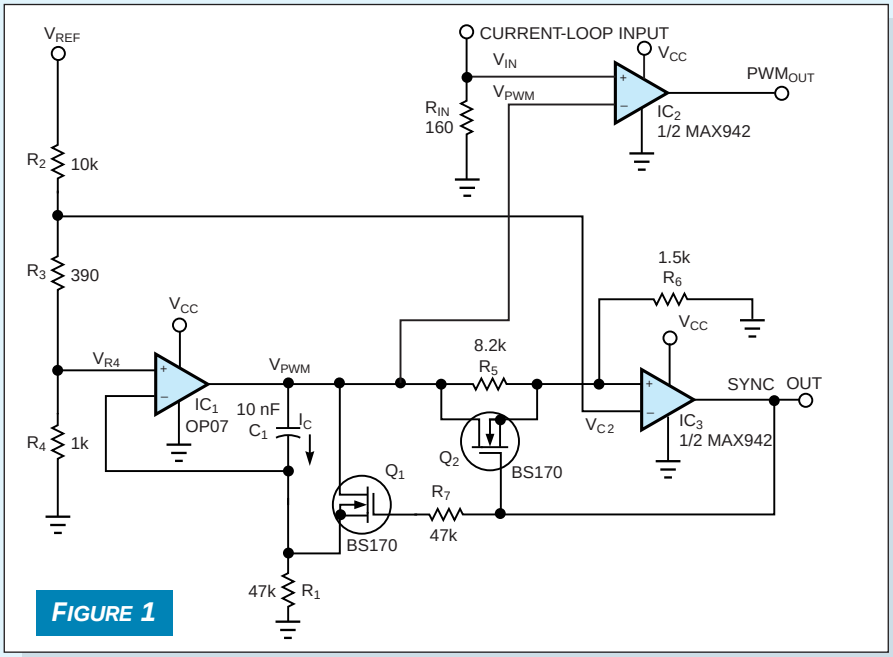


FIGURE 1

A free-running oscillator powers a current-loop controller for PWM signals. Duty cycle varies from 0 to 80% over the full-scale current range.

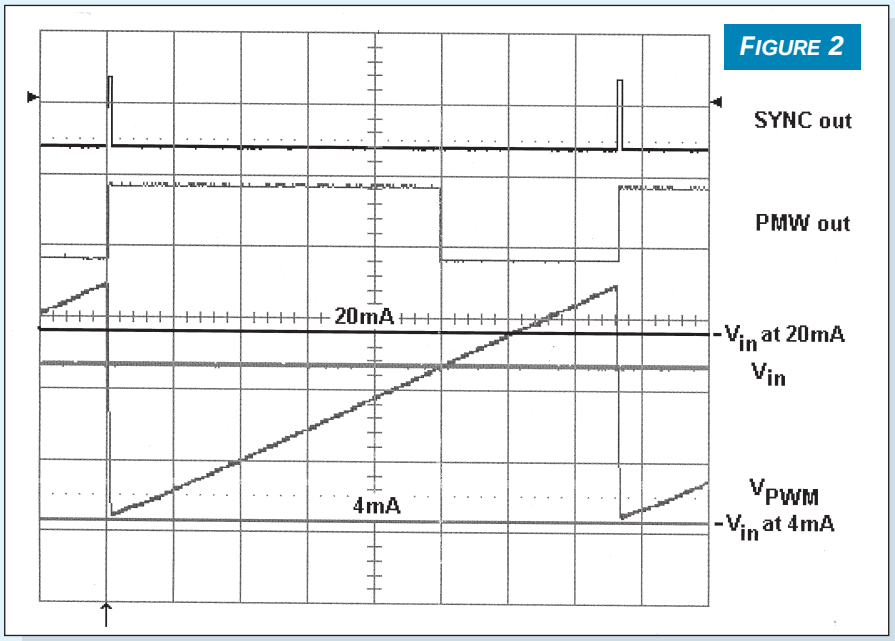


FIGURE 2

This simplified version of the circuit in Figure 1 has a synchronization input instead of an output.

circuit in **Figure 1** to the Sync input of the circuit in **Figure 3**, you can omit R_2, R_5, and C_2. The following formulas apply to **Figure 1**:

$$V_{R4} = V_{REF}\frac{R_4}{R_2+R_3+R_4}.$$

$$V_{PWM(MIN)} = V_{R4}\frac{R_5+R_6}{R_6}.$$

$$I_C = \frac{V_{R4}}{R_1}.$$

$$V_{C-} = V_{REF}\frac{R_3+R_4}{R_2+R_3+R_4}.$$

$$V_{PWM(MAX)} = V_{C-}\frac{R_5+R_6}{R_6}.$$

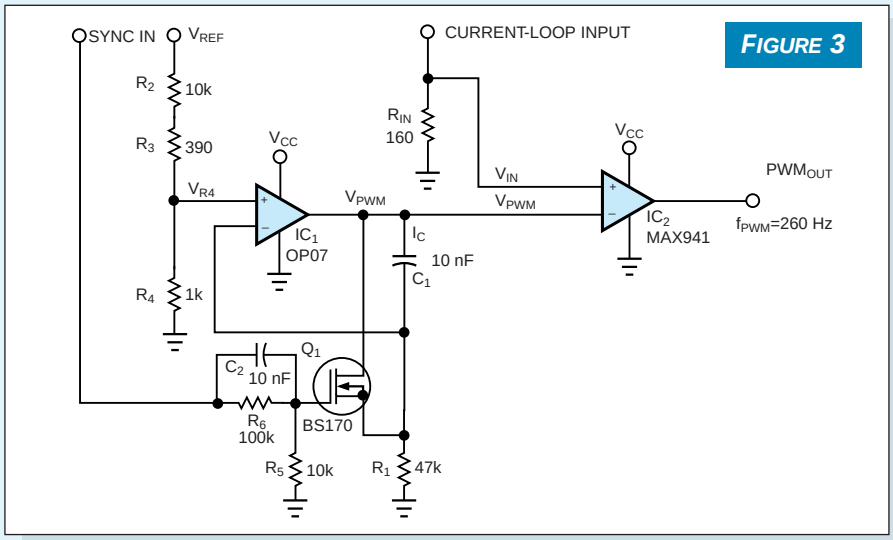
$$f_{PWM} = \frac{I_C}{C_1(V_{PWM(MAX)} - V_{PWM(MIN)})}.$$

These equations apply to **Figure 3**:

$$V_{R4} = V_{REF}\frac{R_4}{R_2+R_3+R_4}.$$

$$V_{PWM(MIN)} = V_{R4}\frac{R_5+R_6}{R_6}.$$

FIGURE 3



The circuit in Figure 1 provides a linear duty-cycle-versus-current function. V_{PWM}, a function of the current, determines the duty cycle.

(DI #2242 e

$$I_C = \frac{V_{R4}}{R_1}.$$

To Vote For This Design, Circle No. 356

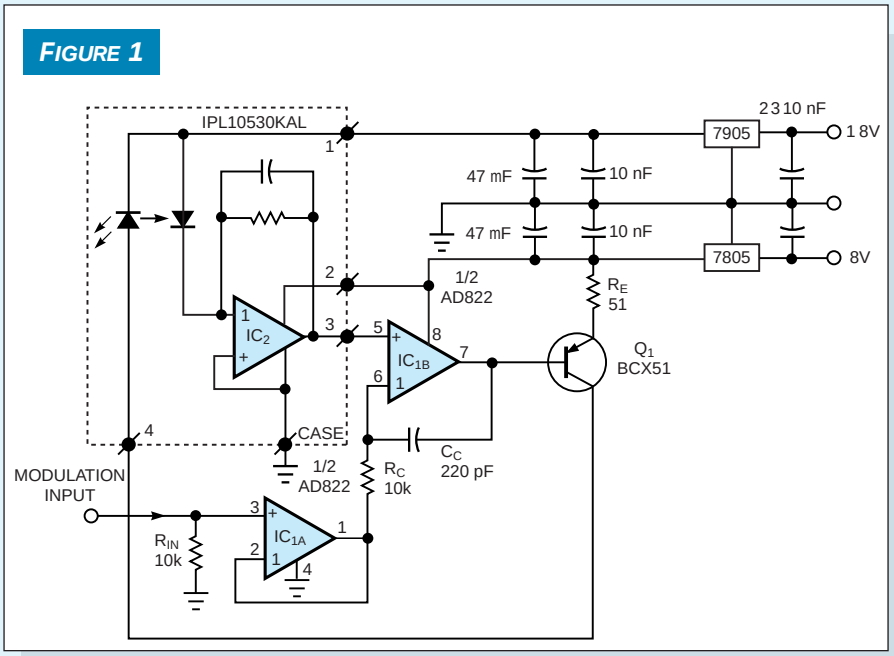
Circuit uses simple LED for near-IR light

LUKASZ ŚLIWCZYŃSKI, MARCIN LIPIŃSKI, INSTITUTE OF ELECTRONICS, KRAKOW, POLAND

You can successfully use LEDs as sources of near-infrared light. However, when you need a source of light with precisely controlled power, a feedback loop is necessary to compensate for the temporal and thermal changes of the LED parameters. Standard LED types come with neither these monitoring detectors nor an external monitoring photodiode to detect part of the emitted light and generate a feedback signal. The situation calls for some mechanical fixture to mount the photodiode. Such a solution, however, is bulky and cumbersome, especially when space is scarce. You can solve the problem by using an 880-nm IPL10530KAL hybrid detector/emitter module from Integrated Photomatrix Ltd. A modulated IR light source uses only one dual op amp, a transistor, and two voltage regulators (**Figure 1**).

The modulation input acts as a reference voltage and connects via amplifier IC_{1A} to the comparing feedback-loop

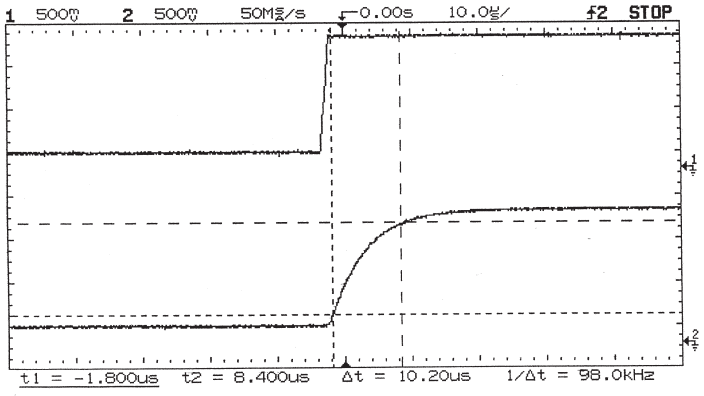
FIGURE 1



A few inexpensive components connected to a detector/emitter module enable you to obtain precise control of near-IR emitted light.

amplifier, IC_{1B}. The resulting output voltage drives the output transistor, Q_1, which directly drives the LED. Resistor R_E limits the maximum drive current to approximately 80 mA, thus preventing damage to the LED. You can reduce R_E's value if you need higher power levels; the absolute maximum rating for the LED is 500 mA. If you change R_E, you may need to alter the frequency-compensation network. The network comprises R_C and C_C and introduces a pole at 0 Hz and a zero at $\frac{1}{2}pR_CC_C$ into the open-loop transfer function. The zero cancels the pole (at approximately 100 kHz) that the monitoring-photodiode preamplifier introduces, so the pole is a dominant pole in the feedback loop.

FIGURE 2



IC_2 in Figure 1 delivers a clean response to a modulation-input step function.

With the component values shown, the 3-dB modulation bandwidth of the source is approximately 40 kHz. You can experimentally determine the value of the compensating capacitor, C_C by observing the voltage at Pin 3 of IC_2 and driving the circuit with a square wave (**Figure 2**). The output is a filtered version of the optical-output waveform. A Spice simulation shows the phase margin to be approximately 85°. The AD822 dual rail-to-rail op amp accommodates modulation-input voltage from 0 to 5V. The slope efficiency of the entire source (defined as dP_I/dV_{MOD}) is approximately 1.5 mW/V and may vary slightly from unit to unit of detector/emitter modules.

You can use this design as an IR light source in a precise reflectrometric measurement system incorporating pulse modulation and synchronous detection. To increase accuracy of the system comprising the entire optical head, you can install the system in a thermally stabilized environment with temperature controlled to within 0.5°C. The long-term measured power stability of the source is better than 1 ppm after initial warm-up. (DI #2243) e

To Vote For This Design, Circle No. 357

Two-DAC circuit adds and subtracts

V MANOHARAN, NAVAL PHYSICAL AND OCEANOGRAPHIC LABORATORY, KOCHI, INDIA

A typical way to add two binary words and provide an analog output is to use several digital ICs that drive a DAC. The circuit in **Figure 1** eliminates the use of several digital-IC packages and, hence, the need for the digital power supply. The circuit simultaneously carries out addition and subtraction on two 8-bit binary words and presents the output in bipolar analog form.

The hardware consists of four ICs, and the operation takes only 85 nsec, which is the settling time of the DACs plus the settling time of the op amp. IC_1, a precision 10V reference, provides the reference current for both multiplying DACs: IC_2 and IC_3. For these DACs, $I_{REFA}=10V/R_1$, and $I_{REFB}=10V/R_2$. In this case, $I_{REFA}=I_{REFB}=I_{REF}=2$ mA.

The output currents, I_{OA} and I_{OB}, depend on the respective A and B binary inputs and the input reference currents as follows:

$$I_{OA} = I_{REF} \bullet \left(\frac{N_A}{2^n}\right), \quad (1)$$

and

$$I_{OB} = I_{REF} \bullet \left(\frac{N_B}{2^n}\right), \quad (2)$$

where n is the number of input bits and N_A and N_B range in value from 0 to 2^n-1, in accordance with the input binary words.

The DAC-08 has complementary current outputs. Therefore, you can express the complements of I_{OA} and I_{OB} as

$$\overline{I_{OA}} = I_{FS} - I_{OA}, \quad (3)$$

and

$$\overline{I_{OB}} = I_{FS} - I_{OB}, \quad (4)$$

where I_{FS}, the full-scale current of the DAC, is

$$I_{FS} = \frac{2^n - 1}{2^n} \bullet I_{REF}. \quad (5)$$

The circuit configures IC_{4B} as a current-to-voltage converter. Thus,

$$V_{OUT(A+B)} = I_{OA} \bullet R_5 + I_{OB} \bullet R_5. \quad (6)$$

Substituting I_{OA} and I_{OB} from **Equations 1** and **2** into **Equation 6** yields

$$V_{OUT(A+B)} = I_{REF} \bullet \left(\frac{N_A}{2^n}\right) \bullet R_5 + I_{REF} \bullet \left(\frac{N_B}{2^n}\right) \bullet R_5 = \frac{I_{REF} \bullet R_5}{2^n} \bullet (N_A + N_B). \quad (7)$$

IC_{4A} serves as both a current-to-voltage converter for I_{OB} and a buffer to the potential drop across R_3 because of the flow of I_{OA}. Thus, assuming $R_3=R_4$,

$$V_{OUT(A-B)} = \overline{I_{OB}} \bullet R_4 - \overline{I_{OA}} \bullet R_3 = (\overline{I_{OB}} - \overline{I_{OA}}) \bullet R_4. \quad (8)$$

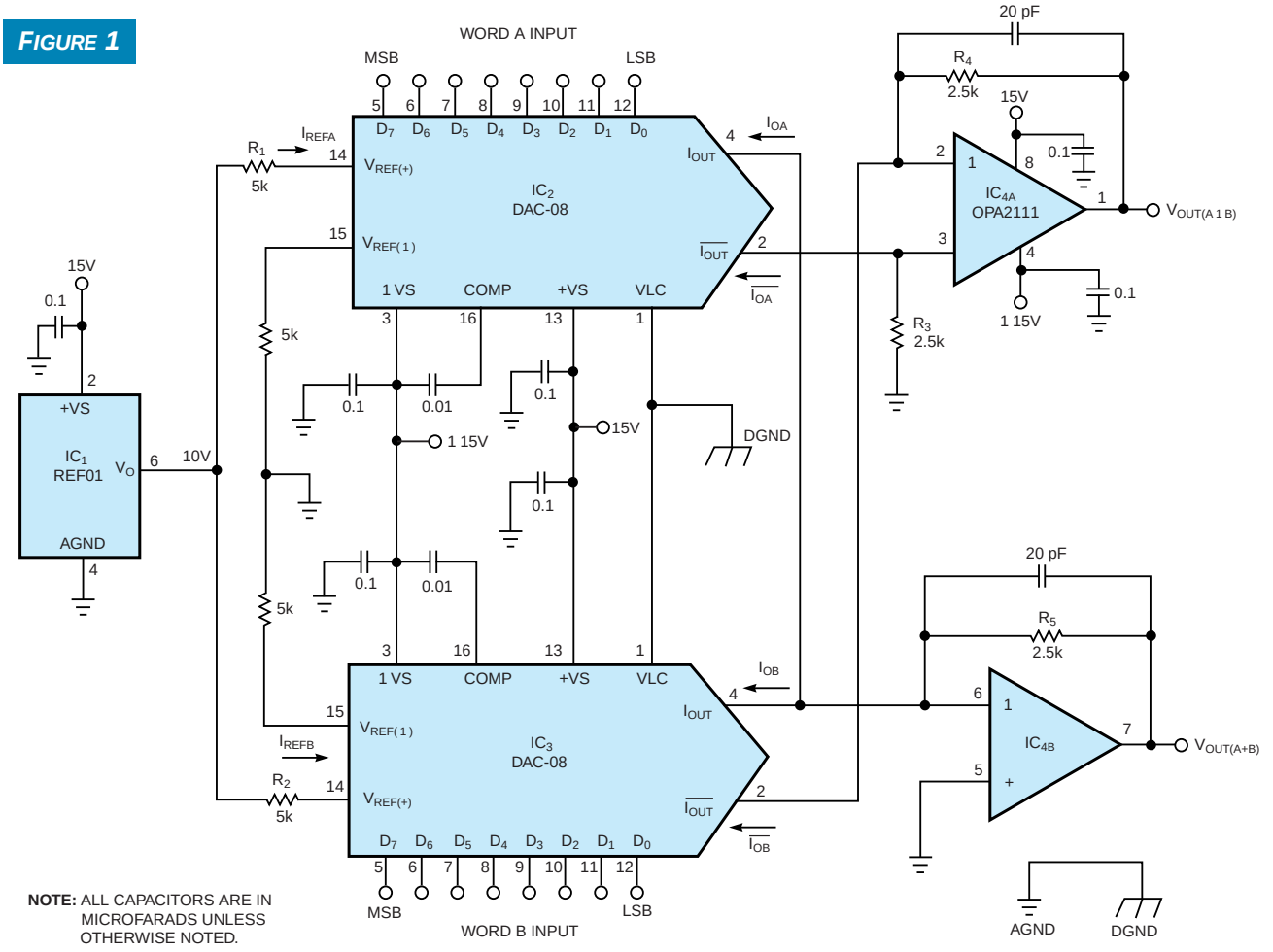
Substituting I_{OA} and I_{OB} from **Equations 3** and **4** into **Equation 8** and assuming that $R_4=R_5$ result in

$$V_{OUT(A-B)} = [(I_{FS} - I_{OB}) - (I_{FS} - I_{OA})] \bullet R_4 = (I_{OA} - I_{OB}) \bullet R_5 = \left[I_{REF} \bullet \left(\frac{N_A}{2^n}\right) - I_{REF} \bullet \left(\frac{N_B}{2^n}\right)\right] \bullet R_5 = \frac{I_{REF} \bullet R_5}{2^n} \bullet (N_A - N_B). \quad (9)$$

(DI #2226) e

To Vote For This Design, Circle No. 358

FIGURE 1



Two DACs and two op amps simultaneously carry out both addition and subtraction on two 8-bit binary words.

Calibration technique uses sound

AUBREY KAGAN, WEIDMULLER LTD, MARKHAM, ON, CANADA

A number of Weidmuller Ltd's (www.weidmuller.com) products require output calibration for zero and span. Normally, a technician calibrates a product by monitoring a digital readout and adjusting the zero and span potentiometers in turn. Because the technician must concentrate so heavily on the readout, the technician often experiences eyestrain, and the screwdriver often slips from the potentiometer. You can use an ADC card to implement the calibration function on a PC, and you can add an audio feature to simplify the calibration. In some cases, the use of audio can replace the need for a digital readout.

The audio spectrum arbitrarily divides into two sections: The first section ranges from 100 to 500 Hz; the second is 1 kHz and higher. The method compares the measured value from the device under test (DUT) with the desired value. If the measured value is lower than the desired value, the output is a frequency from the lower range. Conversely, if the measured value is higher than the desired value, the result is a frequency from the higher range. The further away the measured value is from the desired value, the lower or higher in frequency the audio signal becomes. The object is to adjust the relevant potentiometer to produce no tone—a result that occurs within a window around the desired value. Although the calibration process is intuitive, the program in **Listing 1** visually prompts the technician about which potentiometer to adjust and in which direction to turn it. The technician quickly becomes accustomed to which speed, based on the audio frequency, he or she should adjust the potentiometer to and learns to slow the trimming as the measured value approaches the desired value. The program uses Turbo C++; **Listing 1** shows only the relevant sections of code. (You can download **Listing 1** from *EDN*'s Web site: www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2246.) (DI #2246) e

To Vote For This Design, Circle No. 359

LISTING 1—AUDIO-CALIBRATION ROUTINE

```
#include <dos.h>
#define TOLERANCE 10

int tv,vt.in_range;
int channel;


//prototypes
void soundlo (unsigned int freq);
void soundhi (unsigned int freq);
int ADCreading (int channel);
int FetchSetPoint(void);

void main (void)
{
...
...
...
     tv=ADCreading(channel);
     //reading the A/D channel and conditioning it to the desired units
     vt=FetchSetPoint();
     //fetch the desired value
     in_range=0;
     //set flag for out of range
     if (tv>=vt)
     {//select frequency in upper range
          if ((tv-vt)>TOLERANCE)
          {//make a sound proportional in upper range
               soundhi(tv-vt);
          }
          else
          {//turn off sound
               nosound();
               in_range=1;
          }
     }
     else {
     //select frequency in lower range
          if ((vt-tv)>TOLERANCE)
          {//make a sound proportional in upper range
               soundlo(vt-tv);
          }
          else
          {//turn off sound
               nosound();
               in_range=1;
          }
     }
...
...
...

}

void soundhi (unsigned int freq)
{

     sound(1000+((freq)/2));
}

void soundlo (unsigned int freq)
{
unsigned int temp;
     //checking and limiting minimum frequency to 100Hz
     if ((freq/2)>=400)
          temp=100;

     else
          temp=500-((freq)/2);
     sound(temp);
}□
```

EDITED BY BILL TRAVIS & ANNE WATSON SWAGER

Simple mC acts as dedicated motor control

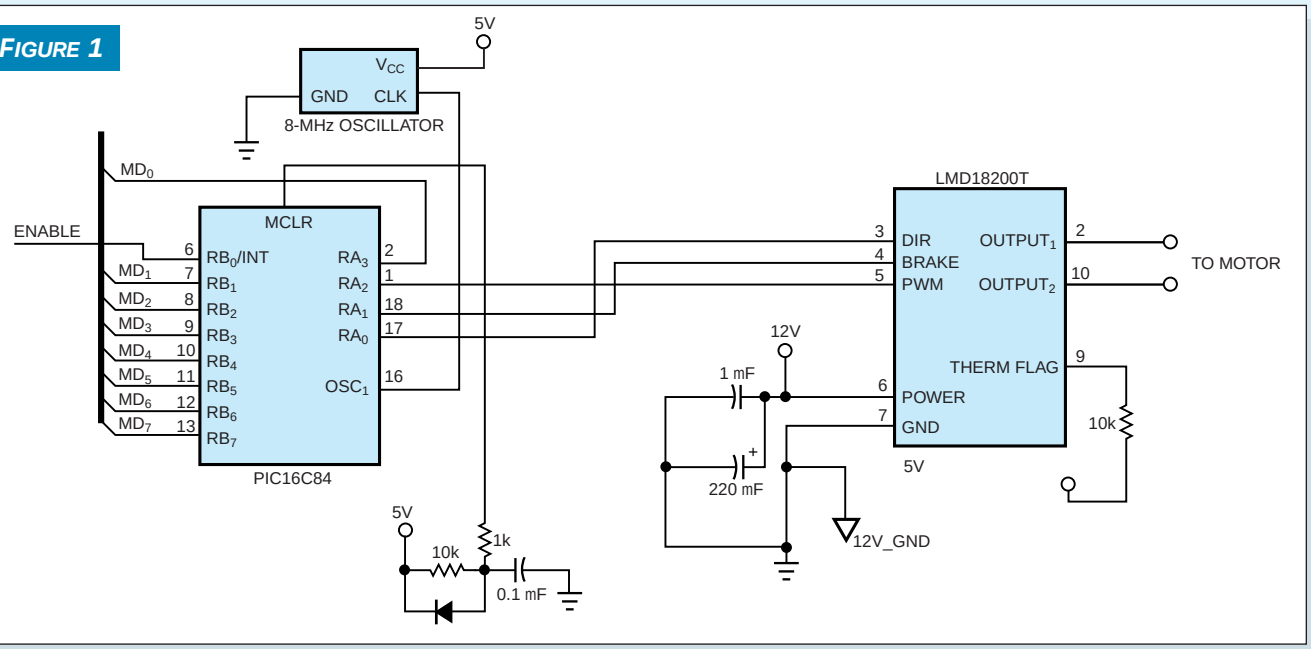
DYLAN HORVATH, GECKO SYSTEMS INC, AUSTIN, TX

Motion-control systems often use a PWM signal to control the duty cycle for a motor driver or amplifier module. Typical designs generate the PWM signals using mCs with dedicated PWM output lines, such as the PIC16C65 (Microchip Technology, www.microchip.com) and the HC11 (Motorola Inc, www.motorola.com). However, these mCs may have more features than necessary for a motion-control system with multiple degrees of freedom. Using this type of mC on each degree of freedom becomes costly, particularly if all you need to do is generate motor-control signals.

An alternative approach uses one low-cost mC, in this case the PIC16C84, as a dedicated motor-control register **(Figure 1)**. The circuit accepts control words from an 8-bit digital bus, and the chip-select line triggers the mC, much the same as other standard 8-bit hardware. You can arrange multiple mCs on a bus and communicate with a higher level motion-control computer or mC. For example, you can use the parallel port of a PC to control all the degrees of freedom on a robot arm. By using PWM signals to modulate the speed at each joint, coordinated motion is possible.

The RA_3 and RB_1-to-RB_7 data-bus lines are digital inputs that connect to an output-controlled data bus. The PIC16C84 ignores these inputs until there is a high-to-low transition on Pin 6 (RB_0/INT). On this transition, the mC places the state of RA_3 on the output RA_0 (direction bit) and places the state of RB_1 on RA_1. The state of the remaining lines, RB_2 to RB_7, set

FIGURE 1



One low-cost mC can operate as a dedicated motor-control register.

TABLE 1—EXAMPLE MOTOR-CONTROL-REGISTER VALUES

Pulse-width-setting bits						BRK,DIR			
D7	D6	D5	D4	D3	D2	D1	D0	HEX	Description
x	X	x	x	x	X	1	x	--	Brake on; motor does not turn
1	1	1	1	1	1	0	1	$FD	Motor turns clockwise, 100% duty cycle
1	1	0	0	1	0	0	0	$C8	Motor turns counterclockwise, ' 80% duty cycle
0	1	1	1	1	1	0	0	$7C	Motor turns counterclockwise, ' 50% duty cycle
0	1	1	0	0	0	0	0	$60	Motor turns counterclockwise, ' 20% duty cycle

the pulse width of the PWM output, RA_2. Note that these outputs are subject to special conditions.

The RB_0/INT digital input latches the word on the 8-bit data bus. A 74HC138 1-of-8 device selector drives this active-low input. The high-to-low transition generates an interrupt to update the state of the register. At all other times, the circuit ignores the state of the 8-bit data bus.

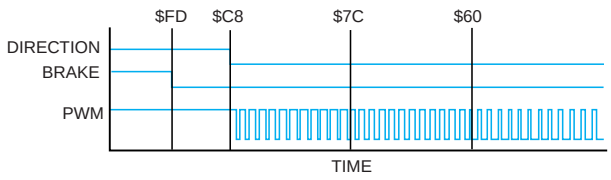
RA_0 drives the motor-drive chip and determines the polarity of the current going through the motor in the output stage of the driver. RA_1 also drives the motor-drive chip. When RA_1 is high and RA_2 is high, active braking of the motor occurs. When RA_1 is high and RA_2 is low, the motor coasts to a stop.

The RA_2 PWM output has a duty cycle that depends on the binary word on the inputs during a high-to-low transition on RB_0/INT. The duty cycle of this signal increases from 1.56 to 100% in increments of 1.56%. In other words, the duty cycle goes from $^1/_{64}$ to $^{64}/_{64}$ in increments of $^1/_{64}$, depending on the binary word on RB_2 to RB_7. The duty cycle repeats at a rate of approximately 300 Hz.

Figure 2 shows the output lines from the motor-control register when you load the values from **Table 1** into the register. During the power-up configuration, the direction, brake, and PWM lines are high. Then, loading the value ($FD) turns off the brake and sets the PWM line at 100% duty cycle (maximum speed). Loading $C8 switches the direction of the motor and reduces the duty cycle to 80%. The next two values ($7C) and ($60) maintain the direction of the motor but reduce its duty cycles to 50% and 20, respectively.

There is a lag between the time the enable line goes active

FIGURE 2



The output lines of the motor-control register change according to the values of the direction, brake, and PWM signals.

low and the time the mC code can read the value on the data bus. When the enable line triggers an interrupt signal, the mC must save the program counter and the state of its internal registers before the mC can process the interrupt. This delay causes a problem if the value on the data bus changes by the time the PIC mC samples it. You can solve this problem by using a latch to store the value on the data bus long enough for the PIC mC to see it.

You can download the corresponding assembly code from *EDN*'s Web site, www.ednmag.com. (At the registered-user area, go into the Software Center to download the file from DI-SIG, #2239.) (DI #2239) e

To Vote For This Design, Circle No. 376

Make a low-cost benchtop power meter

JIM TODSEN, BURR-BROWN CORP, TUCSON, AZ

With a few inexpensive ICs and passive components, you can easily make a multirange power meter suitable for use on your benchtop. The circuit in **Figure 1** measures currents from microamps to amps and voltages as high as 100V. The voltage at V_{OUT}, which you can monitor with a DVM, indicates the load's power. Two 9V batteries can run the circuit (±V=±9V), which has a current drain of 10 mA.

TABLE 1—POWER METER RANGES AND SETTINGS

S_0	S_1	PGA GAIN	I_{MAX}	V_{MAX}	P_{MAX}	V_{OUT} scale
Open	Open	1000	10 mA	100V	50 mW	10 mW/V
Closed	Open	100	100 mA	100V	500 mW	100 mW/V
Open	Closed	10	1A	100V	5W	1W/V
Closed	Closed	1	10A (see note)	100V	50W	10W/V

NOTE: I_{MAX} may be lower, depending on the rating of R_{SENSE}.

The circuit performs an analog multiplication of current and voltage to calculate the power. The load that you want to measure connects between +OUT and –OUT. The supply to the load connects between +IN and –IN. The PGA amplifier (IC_1) produces a voltage proportional to the load current (I_{LOAD}) sensed across R_{SENSE}, which sits on the ground side of the supply. R_1, R_2, and IC_{3D} generate a scaled version of the load voltage equal to $V_{LOAD}/20$. The output of IC_1 and $V_{LOAD}/20$ are the inputs to IC_2's precision analog multiplier. IC_2 has a built-in scale factor of $^1/_{10}$. R_4, R_5, and R_6 provide additional gain. A lowpass filter at the output helps reduce noise and provides protection to IC_2 in case V_{OUT} accidentally shorts to ground. Combining all the scaling factors gives

$$V_{OUT} = (I_{LOAD}R_{SENSE})\left(V_{LOAD}\frac{R_2}{R_1+R_2}\right)PGA_{GAIN}\left(\frac{1}{10}\right)\left(\frac{R_6+R_4}{R_5}\right) = I_{LOAD}V_{LOAD}\frac{PGA_{GAIN}}{10}.$$

The circuit works equally well for positive and negative

load currents and voltages. If the load is producing rather than dissipating power, V_{OUT} reads negative. The scale of V_{OUT} is the same for positive and negative power readings. **Table 1** shows the ranges.

The maximum load-current setting (I_{MAX}) limits the output of IC_1 to 5V to meet head-room requirements when using 9V supplies. D_1 through D_5, R_3, and an LED provide a positive-current-overload warning. When the LED turns on, you should decrease the PGA's gain. A similar string of diodes with opposite polarity can monitor negative-current overloads. Make sure R_{SENSE} has a sufficient rating to handle the maximum current you use. Also, remember that for high I_{LOAD}, there is a significant voltage drop across R_{SENSE}.

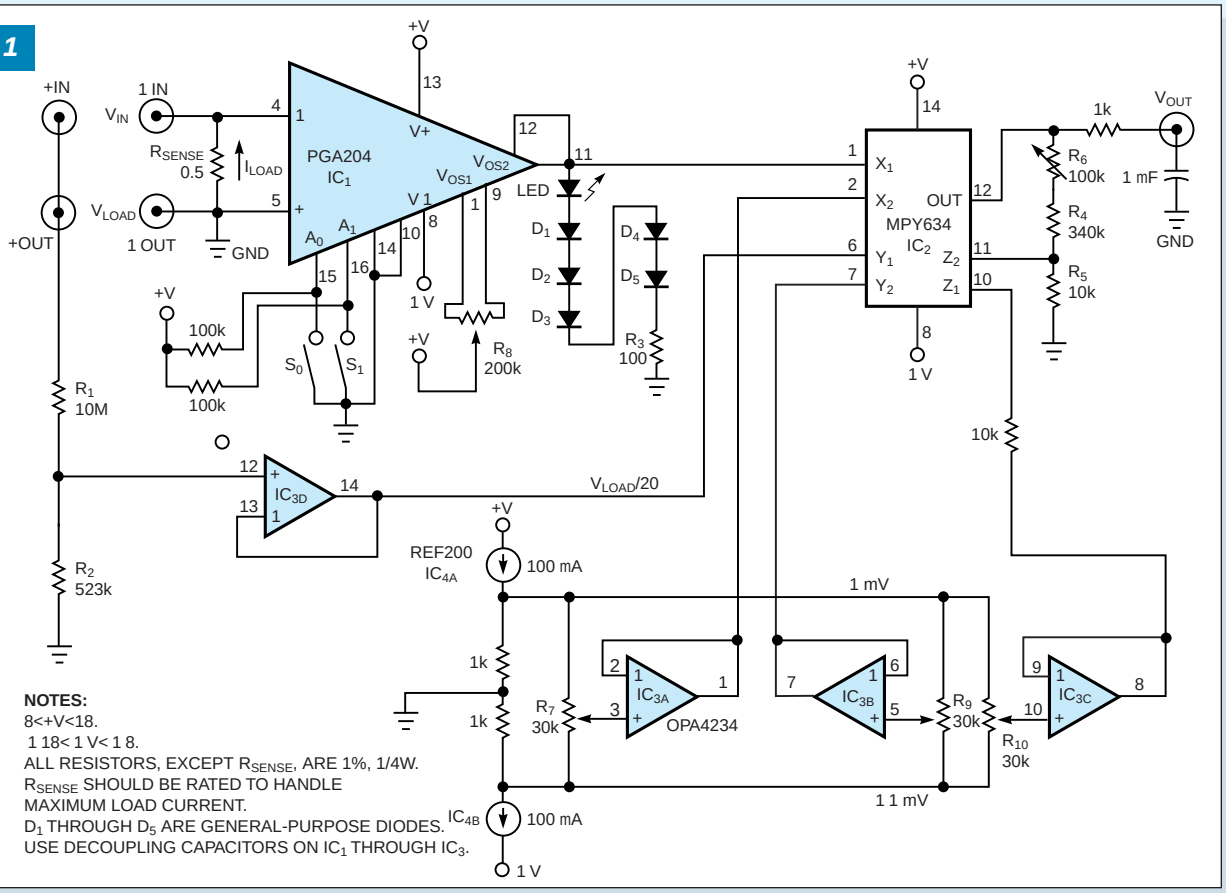
The maximum load voltage (V_{MAX}) of this circuit is 100V, limiting the voltage at IC_2's input to 5V. You can adjust the ratio of R_1 and R_2 for a different V_{MAX}. Keep the sizes of R_1 and R_2 large to minimize current through them. Their currents add to I_{LOAD} and cause an error in the power reading. IC_{3D} prevents IC_2's input-bias current from flowing through R_1 and R_2. The maximum power (P_{MAX}) setting limits IC_2's output to 5V.

IC_{3A} through IC_{3C}, IC_{4A} and IC_{4B}, and potentiometers R_7 through R_{10} provide offset cancellation. R_6 provides gain calibration. The circuit must remove various offsets and gain errors to achieve the best accuracy, which is better than ½% of full-scale over most of the ranges. If lower accuracy is acceptable, you can remove some or all of the offset cancellation circuitry. To fully calibrate the circuit:

1. Short the load (place a short between +OUT and –OUT) with V_{IN}=0. Adjust R_{10} until V_{OUT}=0, which nulls the offset of the output of IC_2.
2. Remove the short, set PGA=1, and apply a large V_{IN} with no load. Adjust R_7 until V_{OUT}=0, which nulls the offset of the I_{LOAD} input to IC_2.
3. Set PGA=1000 and continue applying V_{IN} with no load. Adjust R_8 until V_{OUT}=0, which nulls the offset of the front end of IC_1. If the PGA gain remains the same, R_8 is unnecessary because R_7 cancels the offset.
4. Short the load. Apply V_{IN}, and increase I_{LOAD} until the LED starts to turn on. (For PGA=1000, I_{LOAD} is 10 mA to turn on LED.) Adjust R_9 until V_{OUT}=0, which nulls the offset of the V_{LOAD} input to IC_2.
5. Finally, calibrate the gain. Set the PGA=100, the load=2k, and V_{LOAD}=25V. Adjust R_6 until V_{OUT} matches the calculated power.(DI #2250) e

To Vote For This Design, Circle No. 377

FIGURE 1



A programmable-gain amplifier, an analog multiplier, and a handful of other active and passive components implement a benchtop, multirange power meter.

Low-battery voltage cutoff consumes just 1 mA

YONGPING XIA, TELDATA INC, LOS ANGELES, CA

A low-battery voltage-cutoff circuit prevents overdischarge of a rechargeable battery. An obvious requirement of this circuit is extremely low power consumption. **Figure 1a**'s simple circuit has a measured current consumption of approximately 1.2 mA and uses only two components to perform the low-battery cutoff function for a four-NiCd battery.

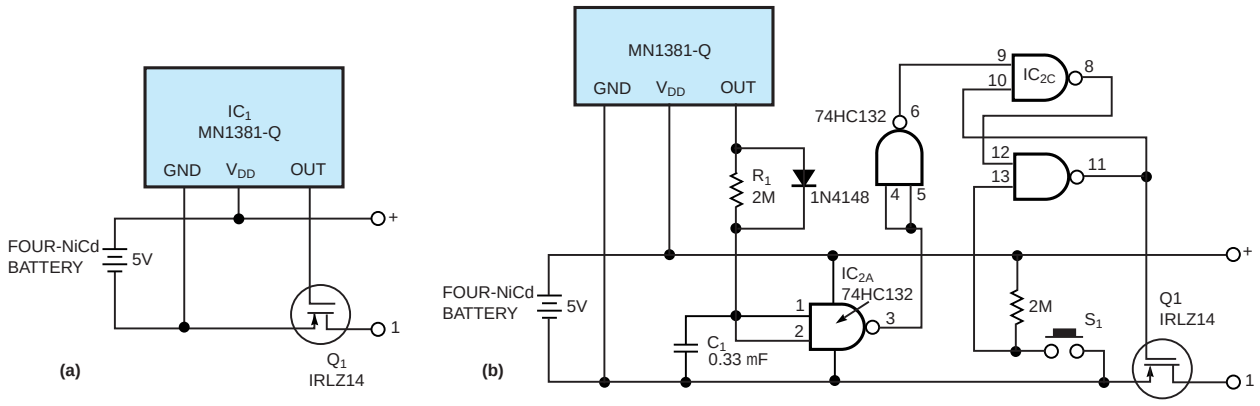
IC_1 is a 3.9V voltage detector with a maximum hysteresis of 0.3V. When the battery is charged, the 5V power supply exceeds this IC's threshold such that its output goes high to turn on Q_1, an IRLZ14 MOSFET switch. The IRLZ14 is a logic-level device with an on-resistance of 0.2V. When the battery voltage drops to below IC_1's threshold, the output of IC_1 is zero, which turns off Q_1.

If the load is heavy, the circuit may turn on and off when the battery voltage reaches the threshold. When the circuit cuts off the load, the battery voltage rises again; this higher voltage may exceed IC_1's turn-on threshold. To prevent this problem, the circuit in **Figure 1b** uses a flip-flop to provide a clean cutoff. Pushing S_1 turns on the switch. When the load has a large capacitance, R_1 and C_1 provide a delayed response to prevent the turn-on in-rush current from triggering the circuit. The power consumption of this circuit is in the same range as that of the circuit pictured in **Figure 1a**.

All the parts for this idea are available from Digi-Key (www.digi-key.com). For a lower switch resistance, you can use the IRLZ44, which has an on-resistance of 0.022V. (DI #2253) e

To Vote For This Design, Circle No. 378

FIGURE 1



These low-battery-detect circuits cut off when the voltage is lower than 4V and consume approximately 1.2 mA.

High-voltage circuit breaker protects to 26V

TED SALAZAR, MAXIM INTEGRATED PRODUCTS, SUNNYVALE, CA

Wide use of the Universal Serial Bus (USB) has led to a selection of overcurrent-protection circuits for supply rails of 2.7 to 5.5V, but few products are available for voltages higher than that range. The circuit breaker in **Figure 1** operates on supply voltages to 26V and trips at a programmed current threshold.

IC_1 is a high-side current-sense amplifier that monitors supply current via the voltage across R_2 and generates a proportional but smaller current at the OUT terminal as follows:

$$I_{OUT} = \frac{R_2 \times I_{TRIP}}{100}.$$

R_1 and R_2 determine the trip current according to the equation,

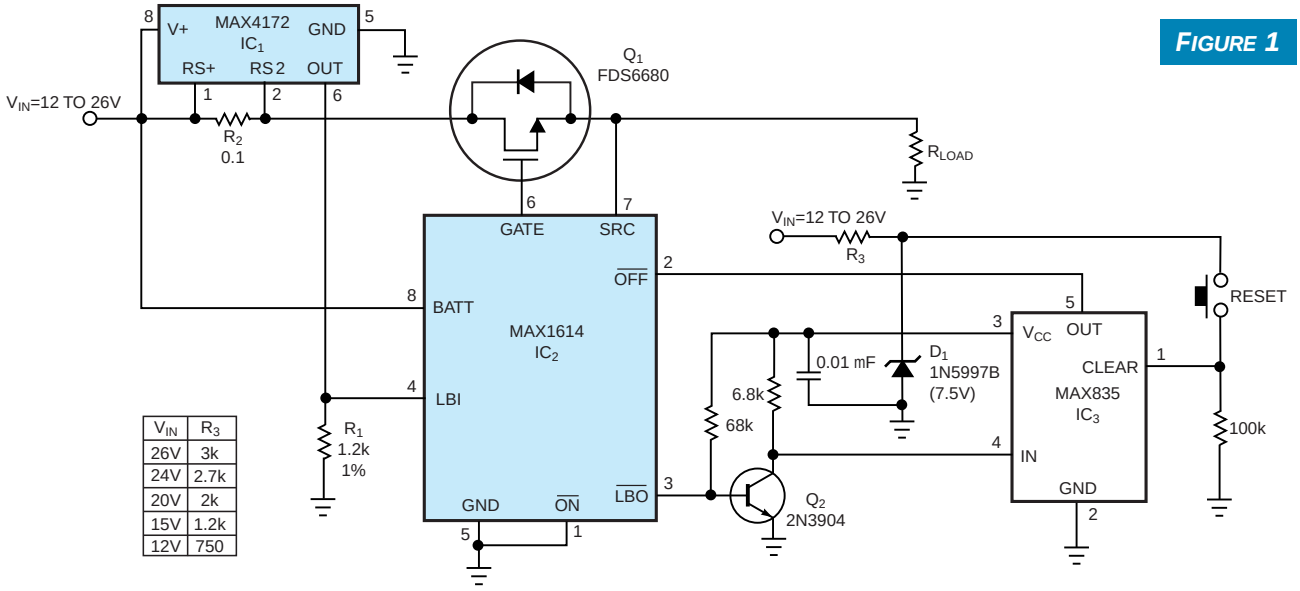
$$R_1 = \frac{120}{R_2 \times I_{TRIP}}.$$

The value of R_1 in **Figure 1** sets the trip current at 1A, but values to 10A are acceptable. Supply current at the trip level produces a voltage across R_1 that triggers the low-battery comparator in IC_2, a high-side, n-channel MOSFET driver. The comparator output ($\overline{LBO}$) drives Q_2 to saturation, causing the latched output of IC_3, a micropower voltage monitor, to go low. Applied to IC_2's Pin 2, this signal disconnects the power by turning off Q_1.

Power remains off until you unlatch IC_2 by depressing the reset button. You may also have to push the button following initial power-up to ensure the correct power-up state. For supply voltages of 12V and higher, choose R_3 according to the table in the **Figure 1**. For supply voltage that is less than 12V, D_1 and R_3 are unnecessary. The signal delay from IC_3 to the load via IC_2 and Q_1 has a turn-off time of approximately 7 msec **(Figure 2a)** and a turn-on time of approximately 400 msec **(Figure 2b)** (DI #2252) e

To Vote For This Design, Circle No. 379

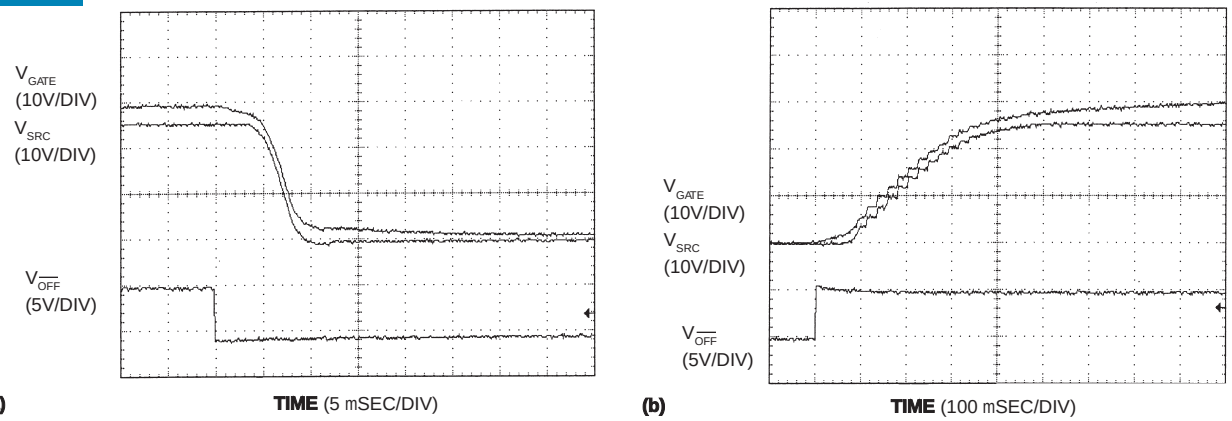
FIGURE 1



NOTE: ALL RESISTORS ARE 5% UNLESS OTHERWISE SPECIFIED.

This circuit provides overcurrent protection for supply-rail voltages to 26V.

FIGURE 2



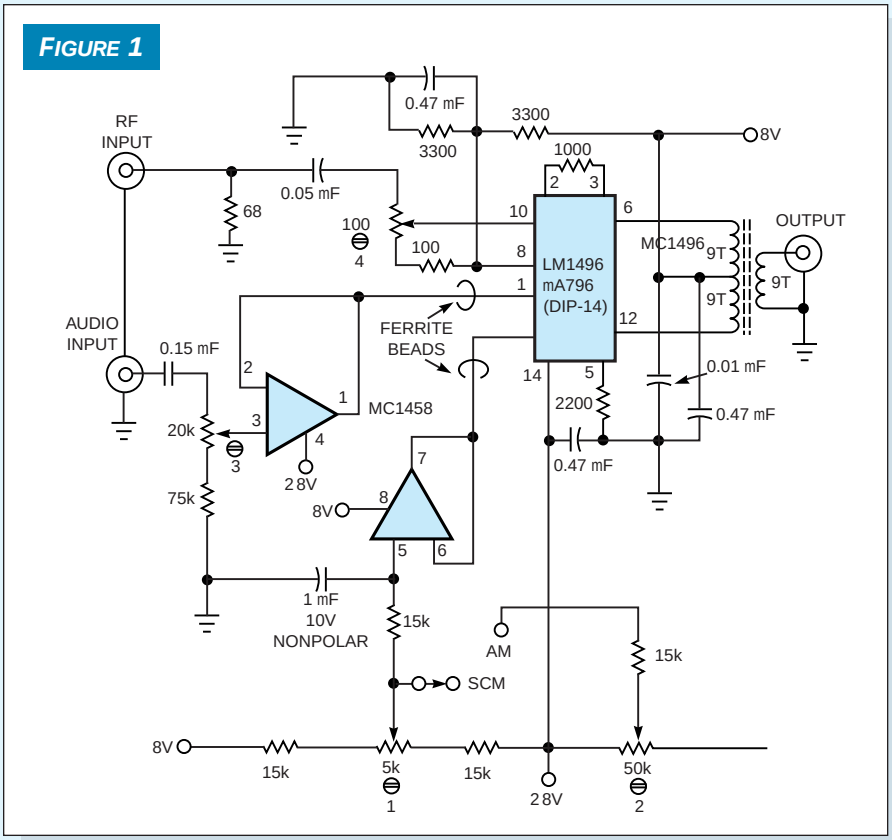
With Figure 1's load-current trip threshold set at 1A, the load voltage, V_{SRC} (middle waveform), turns off (a) and on (b) in approximately 7 and 400 msec, respectively, under the control of the signal at IC_2's V_{OFF} pin.

Add-on modulator has high bandwidth

MJ SALVATI, FLUSHING COMMUNICATIONS, FLUSHING, NY

The simple circuit in **Figure 1** is an add-on modulator that converts the output of a continuous-wave (CW) source to either an amplitude-modulation (AM) or a suppressed-carrier-modulation (SCM) format. Because the circuit has unity gain and 50V input and output impedances, the CW generator's output-level indications remain valid. The frequency response is flat from 0.3 to 45 MHz and only 0.1 dB down at 0.1 and 60 MHz. The modulation bandwidth is similarly broad: flat to 50 kHz and 3 dB down at 15 Hz with the capacitive coupling shown in **Figure 1**. Modulation levels to 100% are possible. Because the modulation sensitivity is 10% per 100 mV rms of modulating signal, you can read the modulation level directly from the audio generator's output-level indicator.

FIGURE 1



A few trimpots, a toroidal transformer, and a dual op-amp interface with a modulator IC to form a linear, high-bandwidth AM modulator.

The circuit is a variation of a standard LM1496/1596 amplitude-modulator setup. It differs from the standard in that it uses a toroidal transformer to provide impedance matching and maximally efficient drive for a low-impedance load, and it drives the modulation ports through unity-gain op amps. The op amp driving Pin 1 provides a high input impedance; thus, it lessens the demands on the audio source and allows practical values for the coupling capacitor. If the audio signal source has no dc component, you can omit the coupling capacitor. You can wind the toroidal transformer with 24-gauge telephone wire over a ferrite core taken from a Sony (www.sony.com) 1-421-302 line choke. A Ferronics (www.ferronics.com) 11-261-J or JW Miller (www.bellind.com) F-50-1 core work equally well. **Figure 1** indicates the adjustment order for the four trim pots. Initially, set all pots to midpoint and inject a 50-mV rms carrier into the RF-input connector. Set the modulation-code switch to SCM and adjust the 5-kV pot for exactly 0V dc at Pin 5 of the MC1458. Next, connect an audio signal to the audio-input connector and switch the modulation mode to AM. Adjust both the 50-kV pot and the audio-signal level until you achieve 100% modulation with no peak clipping and no trough overshoot. Once the biasing is set, set the audio generator's output to exactly 500 mV rms then adjust the 20-kV pot for exactly 50% modulation. Last, set the RF input at exactly 50 mV rms and adjust the 100V pot for 50-mV rms output into a 50V load. (DI #2245) e

To Vote For This Design, Circle No. 380

Simulate signals for telecomm tests

SAMUEL KEREM, PATTON ELECTRONICS, GAITHERSBURG, MD

The circuit in **Figure 1** is a miniature gadget that is helpful in telecommunication applications. The function of the device is to simulate data flow with predefined patterns and use these patterns to check a cable's or a receiver's functionality. The circuit generates a signal in accordance with alternating-mark-inversion (AMI) code. In this type of coding, pulses with alternating polarities represent ones; signals with zero amplitude represent zeros. **Figure 2** shows some examples.

The circuit can produce three AMI-code signal patterns: 1–1–1–1–..., 1–0–1–0–..., and 1–0–0–1–0–0–1–.... As D and E show, in AMI coding, alternating-polarity pulses denote logic one; no pulse denotes logic zero.

The circuit uses a strobe-pulse source, consisting of IC_{1A} and IC_{1B}. The strobe initiates on the falling edge of the clock only if the previous strobe pulse is over. The strobe-pulse duration is a function of R_1C_1. The pulse depends on the state of S_1 and can be nonexistent or close to either 1.5 or 2.5 periods of the clock source (**Figure 2b**). Therefore, the strobe pulse cuts off at? zero, one, or two pulses from the original clock source (**Figure 2c**). IC_2 divides the modified clock frequency by two and restores the duty cycle to 50%. The signal from IC_2 alternatively switches IC_3's internal amplifiers between inverting and noninverting modes with equivalent gain. Thus, IC_3's output is a three-level signal.

R_2, C_2, IC_{4A}, and IC_{4B} introduce a delay of a few nanoseconds to set the internal amplifiers before the clock signal (**Figure 2c**) changes at IC_3's inputs. R_3 through R_6 set the signal level to the appropriate range. You need IC_5 only if your power supply cannot produce ±5V. You calculate the values of R_1 and C_1 for an 8.448-MHz clock source (E2 bit rate). For other clock rates, you must recalculate only C_1's value. (DI #2247) e

To Vote For This Design, Circle No. 381

FIGURE 2

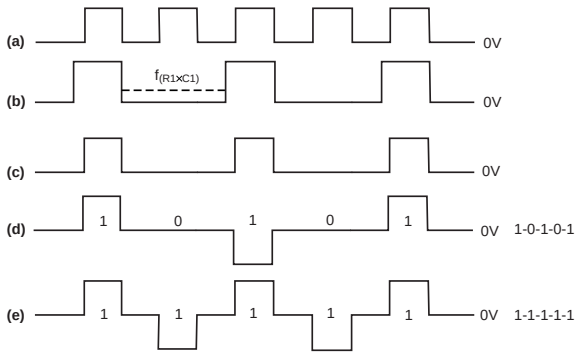
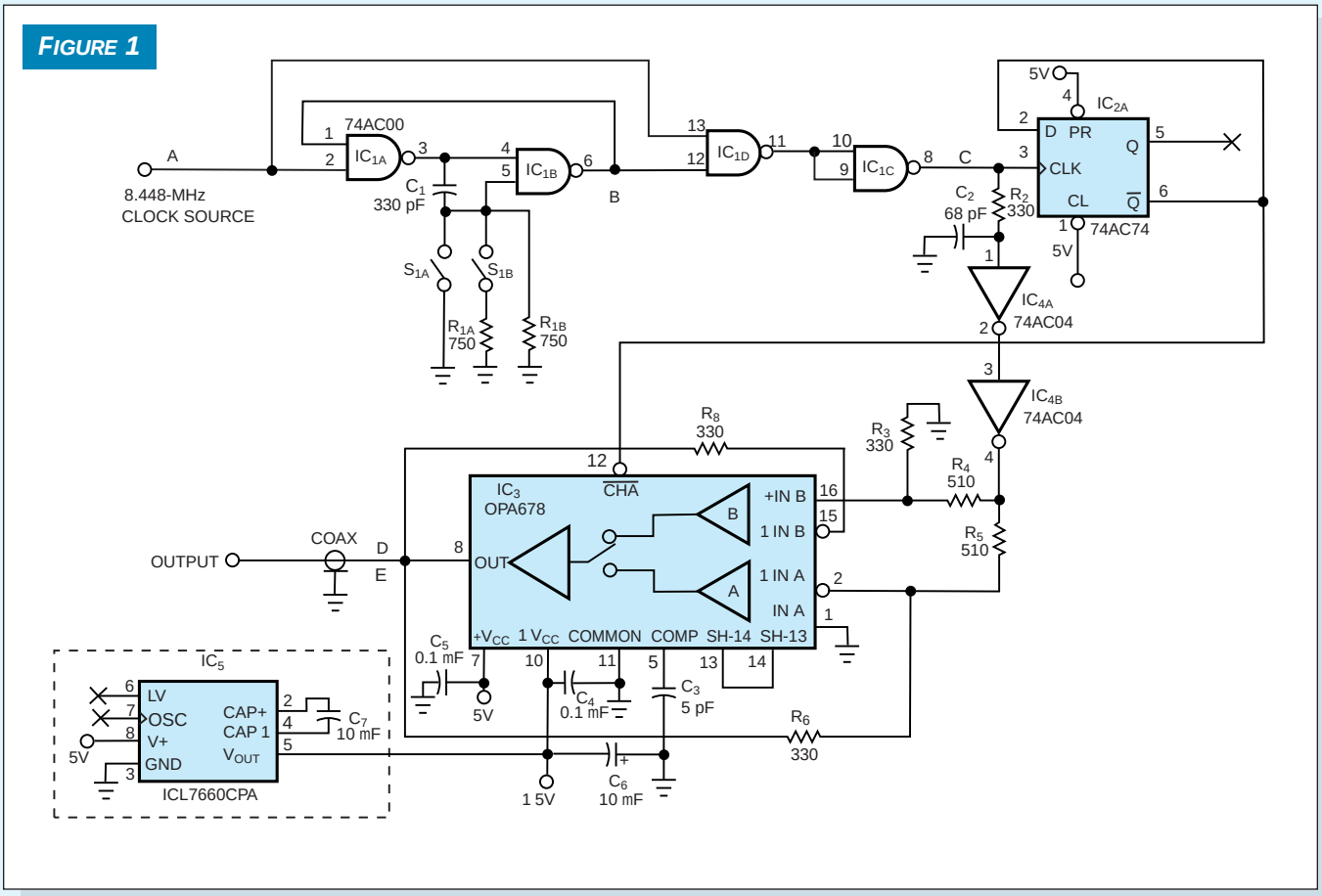


FIGURE 1



A simple bit-pattern generator allows you to test telecommunication equipment.

EDITED BY BILL TRAVIS & ANNE WATSON SWAGER

Bipolars provide stable current source

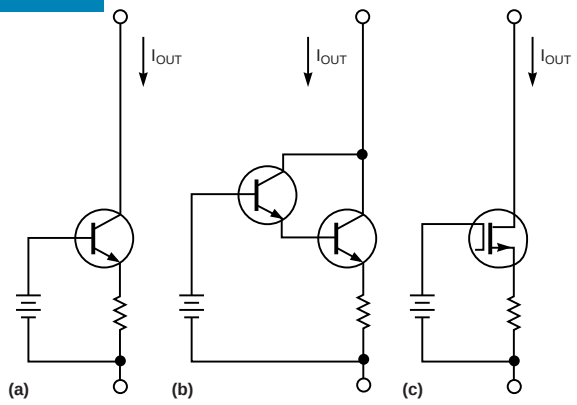
BILL MORONG, MORONG'S HARNESS, DOVER-FOXCROFT, ME

It's possible to implement a precise current source with a useful output at high frequencies, without using operational amplifiers. The circuit in **Figure 1a** suffers inaccuracies from both the V_{BE} drop and the finite base current of the transistor. The circuit in **Figure 1b** overcomes the base-current problem, but has two V_{BE} drops and does not perform well at high frequencies. The circuit in **Figure 1c** has no base-current problem and performs well at high frequencies, but is prone to inaccuracies from the V_{GS} of the FET. The circuit in **Figure 2** largely overcomes these problems.

The V_{BE} of Q_2 cancels that of Q_1. Because the base current of Q_1 diverts (via Q_3) as shown around the current-setting resistor R_1, I_{OUT} is simply two-thirds of V_{IN} divided by R_1. Because the circuit provides error cancellation, the values and voltages are not critical, provided you match the upper and lower components. In this example, Q_1 has a beta of 50; the circuit self-adjusts as beta varies. Neon-driver transistors are a good choice for Q_1 because Q_2 and Q_3 need neither good high-frequency performance nor high output impedance. High-beta (400 in **Figure 2**) transistors are appropriate, as they minimize errors. Insofar as possible, it is desirable to have similar, high betas for Q_2 and Q_3. At high frequencies, it may be beneficial to bypass the base of Q_1 to ground. (DI #2257). e

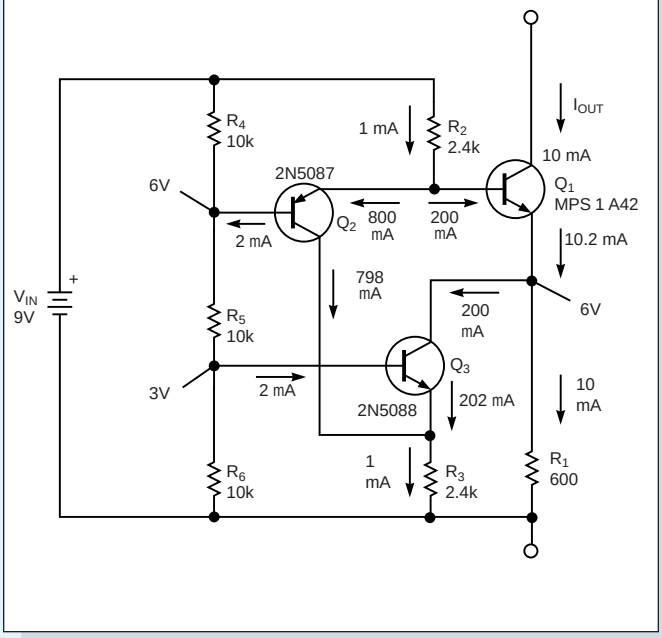
To Vote For This Design, Circle No. 386

FIGURE 1



Finite base current, temperature-dependent base-emitter drop, and gate-source-voltage variations lead to errors in single-transistor current sources.

FIGURE 2



Base-emitter-voltage and base-current cancellation are the keys to a stable, predictable current source.

8-pin µC forms one-chip programmable VCO

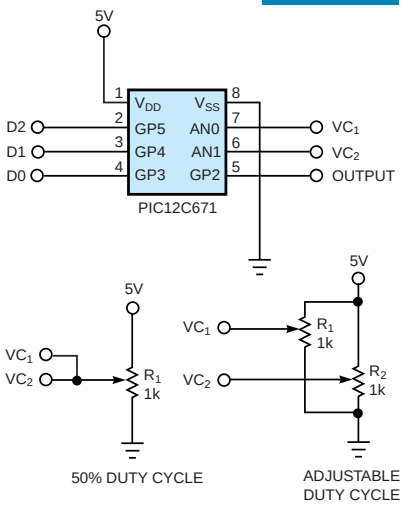
YONGPING XIA, TELDATA INC, LOS ANGELES, CA

The circuit in **Figure 1** uses a Microchip 8-pin µC (PIC12C671) as a voltage-controlled oscillator (VCO). Because the PIC12C671 has an internal 4-MHz oscillator, four-channel 8-bit A/D converters, and built-in power-reset circuitry, you need no external components to configure the VCO. The µC reads two analog inputs through AN0 and AN1. The reference voltage for the A/D conversion is the µC's power supply V_{DD}. The converted 8-bit data determines the duration of output high and output low. Assume, for example, the digitized outputs from AN0 and AN1 are 43 and 87, respectively. Timer 0 loads the 43 after the µC sets output GP2 to logic one. Timer 0 receives its timing from the internal clock.

TABLE 1—TEST RESULTS FOR PIC-BASED VCO

VC1/VC2	D2/D1/D0							
	000	001	010	011	100	101	110	111
0V	899	474	243	123	62.3	31.2	15.6	7.9
0.2V	928	490	252	127	64.4	32.3	16.2	8.1
0.4V	963	509	263	133	67.2	33.8	16.9	8.4
0.6V	1000	531	274	140	70.1	35.2	17.6	8.8
:	:	:	:	:	:	:	:	:
1.0V	1090	579	300	153	77.3	38.7	19.5	9.7
:	:	:	:	:	:	:	:	:
2.4V	1560	858	451	232	118	59.5	29.7	14.9
:	:	:	:	:	:	:	:	:
4.4V	4410	2980	1810	1010	537	268	141	71.1
4.6V	5320	3880	2420	1410	768	402	206	104
4.8V	6720	5550	4130	2730	1470	807	474	243
5.0V	8220	7970	7490	6710	5550	4130	2720	1620

FIGURE 1



A PIC µC makes a handy voltage-controlled oscillator, configurable for either 50% or user-selectable duty cycle.

Once Timer 0 times out, the µC sets GP2 to logic zero before loading 87 into Timer 0. When Timer 0 times out again, the program starts the loop again. Thus, the voltage on AN0 determines the output-high duration, and the voltage on AN1 determines the output-low duration. If a simple 50%-duty-cycle output is satisfactory, you can tie AN0 and AN1 together and control them with one source. You can further program the VCO's output frequency by using D0 through D2. The µC has a prescaler between its clock and Timer 0. Each time the µC reads AN0 and AN1, it also reads D0 through D2 and loads the reading to the prescaler.

Table 1 shows some test results. The frequency range is 8 Hz to 8 kHz. **Listing 1** is the assembly code for the µC. You can download the code from *EDN*'s Web site www.ednmag.com. At the registered-user area, go into the "Software Center" to download the files from DI-SIG, #2259. Because the voltage-to-frequency relationship is purely software-controlled, you can alter the program or use a look-up table to obtain a desired v-f curve. (DI #2259). e

To Vote For This Design, Circle No. 387

ASSEMBLY CODE FOR PIC12C671 VCO

```
TMR0            equ     0x01
STATUS          equ     0x03
GPIO            equ     0x05
INTCON          equ     0x0b
PIR1            equ     0x0c
ADRES           equ     0x1e
ADCON0          equ     0x1f
ADCON1          equ     0x9f
OPT             equ     0x81
port_dir        equ     0x85
high_data       equ     0x21
low_data        equ     0x22
out_status      equ     0x23
temp            equ     0x24

                org     0x0             ;
                goto    main            ;
                                        ;
                org     0x4             ;
                movf    GPIO, 0         ;
                movwf   temp            ;
                rrf     temp, 1         ;
                rrf     temp, 1         ;
                rrf     temp, 1         ;
                movf    temp, 0         ;
                andlw   0x07            ;
                addlw   0x80            ;
                option                  ;
                btfss   out_status, 0   ;
                goto    set_high        ;
                goto    set_low         ;
set_high                                ;
                bsf     out_status, 0   ;
                bsf     GPIO, 2         ;
                movwf   ADCON0          ;
                call    delay           ; data acquisition
                bsf     ADCON0, 2       ; start A/D conversion
loop_2          btfsc   ADCON0, 2       ;
                goto    loop_2          ;
                movf    ADRES, 0        ;
                movwf   TMR0            ;
                bcf     INTCON, 2       ; clear TMR0 interrupt flag
                retfie                  ;
                                        ;
delay                                   ;
                movlw   0x03            ;
                movwf   temp            ;
dly_lp          decfsz  temp, 1         ;
                goto    dly_lp          ;
                return                  ;
                                        ;
main                                    ;
                clrf    GPIO            ;
                bsf     STATUS, 5       ; bank 1
                movlw   0x85            ;
                movwf   OPT             ;
                movlw   0x0b            ;
                movwf   port_dir        ;
                movlw   0x04            ;
                movwf   ADCON1          ;
                bcf     STATUS, 5       ; bank 0
                movlw   0xa0            ;
                movwf   INTCON          ;
loop                                    ; endless loop
                goto    loop            ;
                                        ;
                end
```

Cable tester is fast and cheap

ABEL RAYNUS, ARMATRON INTERNATIONAL, MELROSE, MA

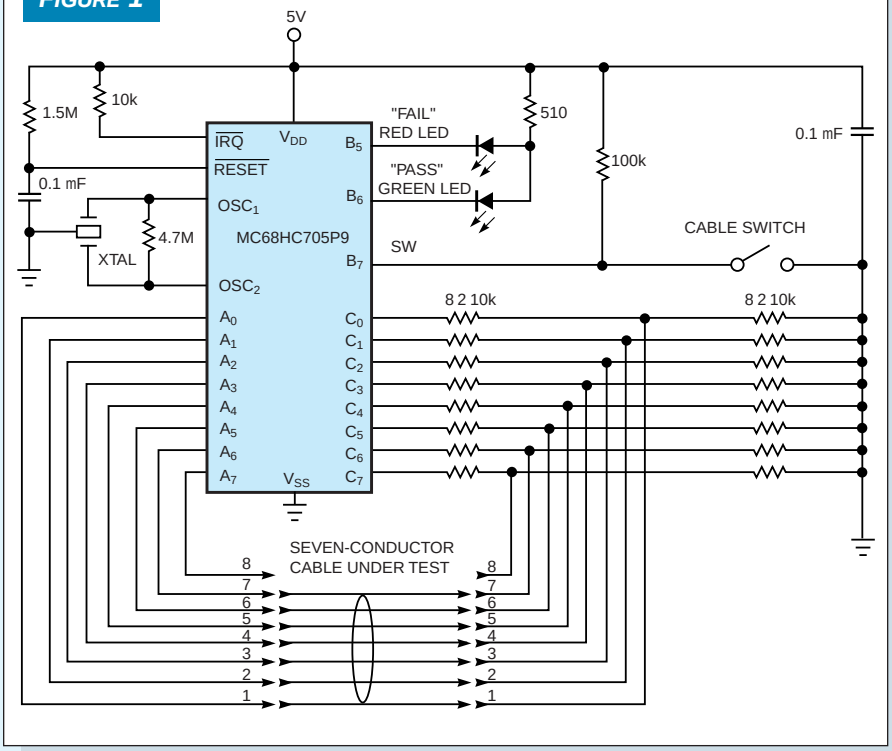
The cable tester in **Figure 1** uses a low-end 8-bit mC. The specific mC to use depends on the number of conductors in the cable you want to test. For the current application, two types of cables were under test, one with three conductors and another with seven. So, the Motorola 68HC705P9 mC was suitable. The program first determines which type of cable is under test by checking the cable-switch position (**Listing 1**). Then, the program checks each conductor line from A_0/C_0 to A_7/C_7 by putting a high-level voltage from output port A on one end of the wire and measuring the response on the other end, which is connected to the input port C. If all of the checks show conductivity, the green "pass" LED turns on. In the opposite case, the red "fail" LED turns on. The test checks not only for conductivity but also for the presence of a cross connection.

If you want to test a variety of cables, you can use more switches. If a cable has more than eight conductors, you can use a different type of mC or multiplex the inputs.

Listing 1 and an assembly-language program are available for downloading from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2225.(DI #2225)

e

FIGURE 1



This simple cable tester verifies that the signal from port A appears at port C of the mC.

To Vote For This Design, Circle No. 388

LISTING 1—SPICE MODEL FOR DEAD-TIME GENERATOR

```
.nolist
$include "STD_P9.asm"
.list
*  I/O PORTS
SW    equ   7 ;prtB
GRN   equ   6 ;prtB
RED   equ   5 ;prtB
*  CONSTANTS
W1    equ   7 ;7 conductors cable
W2    equ   3 ;3 conductors cable
*  VARIABLES
      org   RAM
W     rmb   1 ;wire register
N     rmb   1 ;wire counter
*  INITIALIZATION
      org   MOR
      fcb   $00 ;no watchdog
      org   ROM
init  lda   #%11111111
      sta   ddrA
      lda   #%01111111
      sta   ddrB
      clr   prtC
main  brclr SW,prtB,m1 ;which cable is under test?
      lda   #W1
      sta   W
m2    clr   N
      ldx   #1
start inc   N
      stx   prtA
      txa
      eor   prtC
      bne   FAIL
      lda   N
      cmp   W
      beq   PASS
      aslx
      bra   start
m1    lda   #W2
      sta   W
      bra   m2
FAIL  bset  GRN,prtB  ;green LED off
      bclr  RED,prtB  ;red LED on
      bra   main
PASS  bclr  GRN,prtB  ;green LED on
      bset  RED,prtB  ;red LED off
      bra   main
      org   $1ffe
      fdb   init      ;restart address
.end
.nolist
```

Capacitive sensor "likes" parasitics

BORIS KHAYKIN, CANDID LOGIC INC, MADISON HEIGHTS, MI

Stray capacitance is a common problem with capacitive sensors. The capacitance changes within the measurement range are normally much smaller than the strays; the result is a loss of sensitivity. Various methods are available to increase the relative sensitivity (Df/f_0): frequency subtraction, the use of bridges, and the use of a negatron to subtract the strays, for example. The idea here is not to do battle with the stray, but rather use it and turn its drawbacks to your advantage. This method uses frequency-dependent hysteresis in a classic op-amp multivibrator. **Figure 1** shows a simple, flexible design for a capacitive sensor.

Without capacitor C_2, the design is a classic multivibrator based on comparator IC_1 with output buffer IC_2. If R_1=R_2, the frequency is

$$f = \frac{1}{2R_4C_1 \ln\left(1+\frac{R_2}{R_3}\right)}.$$

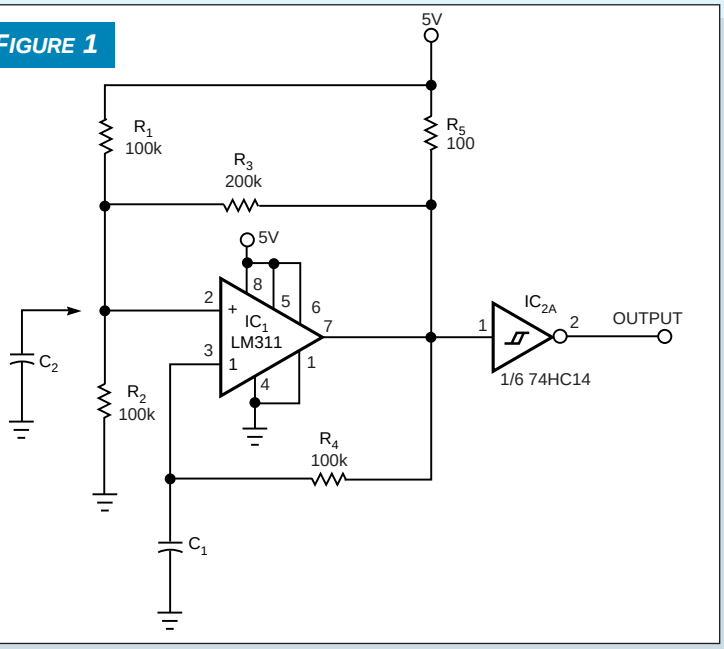
R_1, R_2, and R_3 define the hysteresis, 900 mV with the values shown. Frequency (f) is a function of capacitor C_1, as **Figure 2** shows. Without C_2 and with C_1=60 pF and DC=20 pF, f_0=159 kHz and the relative sensitivity Df/f_0 is –18%. With C_2 connected in parallel with R_2, the hysteresis becomes frequency-dependent. The capacitive reactance (X_C=1/2pfC) in parallel with R_2 reduces the hysteresis in an inverse proportion to the frequency. As a result, the frequency increases. This increase reduces X_C, further reduces the hysteresis, and leads to a further increase in frequency. Thus the relative sensitivity Df/f_0 increases significantly (see **Figure 2** with C_2=40 pF).

With C_2=40 pF, C_1=60 pF, and DC=20 pF, f_0=945.5 kHz and the relative sensitivity (Df/f_0) is –82%. The sensitivity (Df/f_0) (38.6) in this case is 26 times as high as the case without C_2 (Df/f_0=1.45). You can obtain even more interesting results by replacing C_2 with a sensing capacitor. If C_1=200 pF, changing the value of C_2 from 0 to 200 pF changes the hysteresis from 900 to 28 mV, and changes the frequency from 30 to 1300 kHz. **Figure 3** shows output frequency (f) as a function of capacitance C_2. With C_2=100 pF and DC=20 pF, f_0=145.2 kHz and the relative sensitivity (Df/f_0) is +393%. Thus, the frequency is directly proportional to the capacitance.

As **Figure 3** demonstrates, you can adjust the desired initial frequency with R_3, and the sensitivity with R_5. Note that the higher sensitivity in this

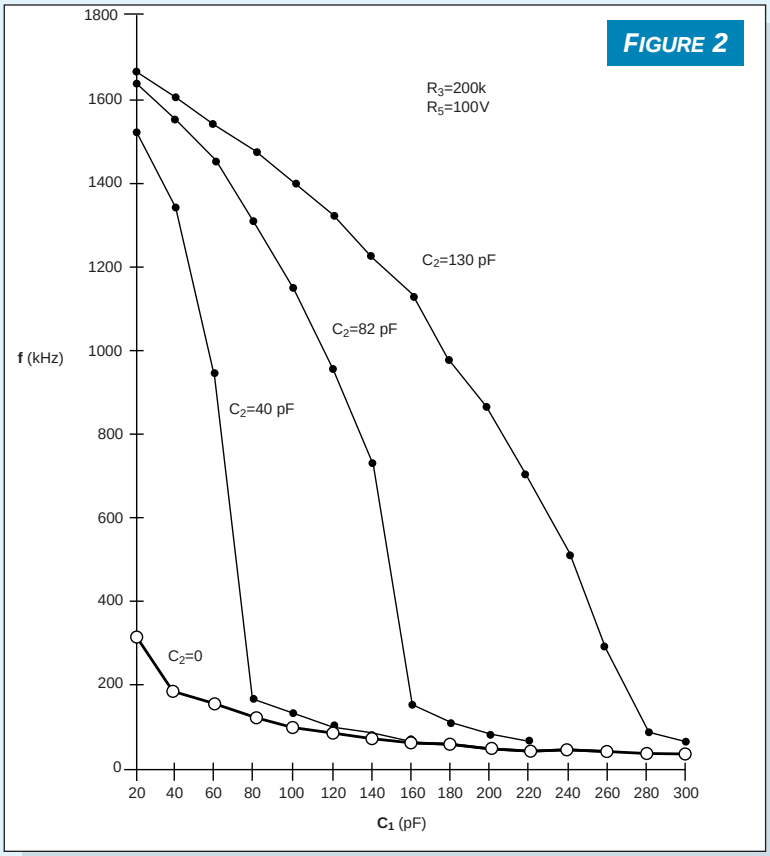
The addition of "stray" capacitance to Figure 1's circuit significantly increases the sensor's sensitivity.

FIGURE 1



If you can't beat 'em, join 'em. This circuit exploits stray capacitance to increase its own sensitivity in making capacitance measurements.

FIGURE 2

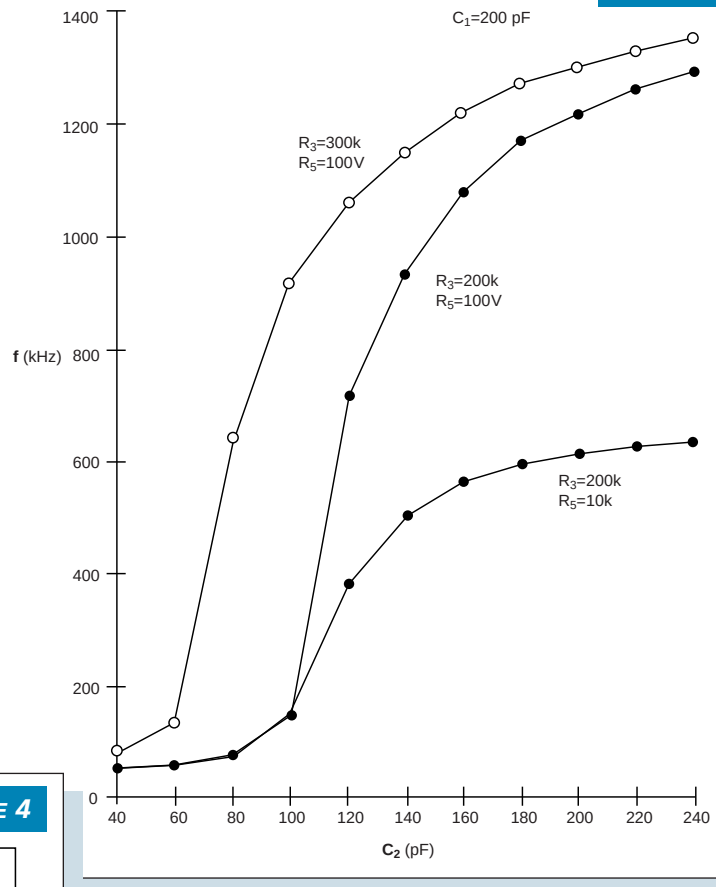


example occurs with a significant stray capacitance (100 pF, for example). If the real sensor has lower initial capacitance (50 pF, for example) the simple addition of a 50-pF capacitor in parallel with the sensor increases the sensitivity. The sensor "likes" the stray capacitance as it produces frequency-dependent hysteresis that, in turn, provides higher sensitivity. You could also use the added capacitor for temperature compensation.

If you use an extremely fast op amp or comparator in this design, there is a certain value of C_2 for which the output frequency jumps up a few kHz with a hysteresis of 5 to 7 pF (**Figure 4**). This quirk is particularly useful in the design of supersensitive capacitive switches. You can adjust the switching point with R_3 and/or a capacitor in parallel with C_2. You can adjust the hysteresis by using a small resistance connected in series with C_2. On the other hand, the use of a slower comparator linearizes the frequency-versus-capacitance characteristic. For example, test results show that with an LM319 comparator, R_3=200 kV, R_5=200V, and C_1=200 pF, the output frequency follows the empirical equation f=140+3.327(C_2–100) kHz with 3% nonlinearity within the range C_2=100 to 400 pF. (DI #2258). e

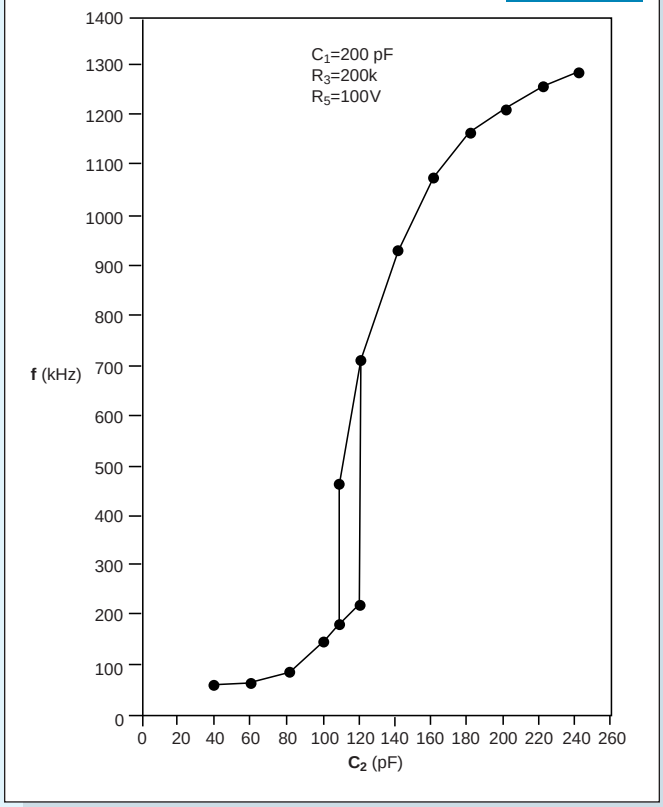
To Vote For This Design, Circle No. 389

FIGURE 3



This response is the result of using the "stray" capacitance as the sensing element in the Figure 1's circuit.

FIGURE 4



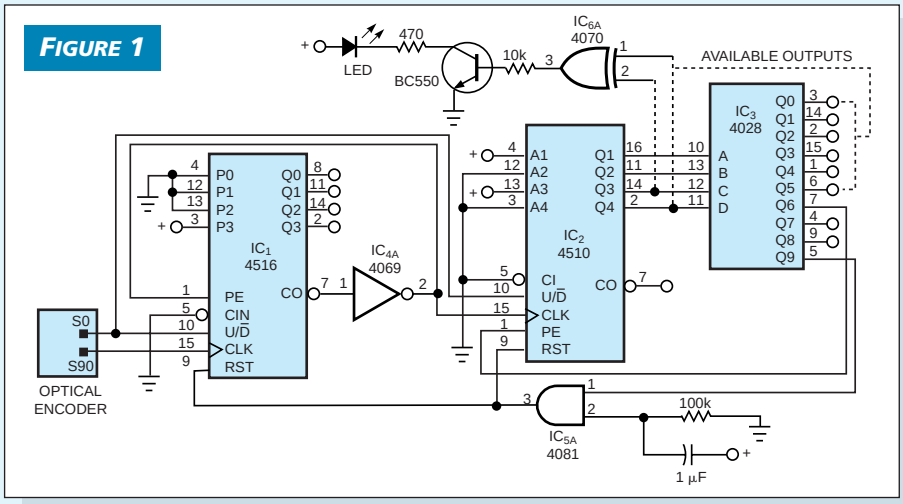
An idiosyncrasy inherent in fast op amps or comparators produces an abrupt jump in frequency for a small change in capacitance.

Optical encoder controls range switch

W DIJKSTRA, WAALRE, THE NETHERLANDS

Instead of using a counter-controlled, pushbutton-activated range switch, you can use an optical encoder. Inexpensive encoders are available, and they occupy minimal space on the front panel of an instrument. Moreover, an encoder gives you the opportunity to select the optimum operating speed. However, at positions near the transition points in counter position, mechanical shocks can provoke false switching. The circuit in **Figure 1** overcomes the false-switching problem.

FIGURE 1



An optical encoder, immune to false switching, takes the place of a counter-controlled range switch.

Output S0 of the optical encoder controls the up/down inputs of counters IC_1 and IC_2. Output S90 connects to the clock input of the HEF4516 binary counter (IC_1). When this counter reaches 0 or 15, the output CO goes low and clocks (via an inverter) the HEF4510 decimal counter (IC_2). Simultaneously, the binary counter assumes a value of eight. Thus, it requires eight pulses of the optical encoder to alter the position of the decimal counter. You must stop turning the optical encoder within eight pulses, which in practice is eminently possible. When you want more security, you can feed the outputs Q3 and Q4 of the decimal counter to an exclusive OR gate. When the output of the XOR gate is high, changing the state of the decimal counter requires a minimum of four pulses from the optical encoder.

The circuit provides control with bidirectional hysteresis. To stop the decimal counter at zero when counting down, the counter resets when output Q9 of the 1-of-10 decoder HEF4028 (IC_3) goes high. To limit the number of decoder positions, you can load the decimal counter one position lower than the maximum output position you want to reach with the output decoder. The configuration in **Figure 1** loads the decimal counter with the value five when Q6 (the seventh position) of IC_3 goes high. (DI #2255). EDN

To Vote For This Design, Circle No. 399

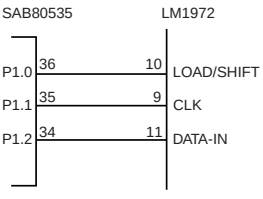
µC controls digital potentiometer

LUKASZ SLIWCZYNSKI, UNIVERSITY OF MINING AND METALLURGY, KRAKOW, POLAND

Many digitally controlled potentiometers (for example, the LM1971/2/3 from National Semiconductor, www.national.com) incorporate a three-wire serial digital interface, using data, clock, and enable lines. In **Figure 1**, the potentiometer's nomenclature for these lines is Data-In, Clk, and Load/Shift, respectively. The assembler program in **Listing 1** provides an interface to an SAB80535 µC. The main idea of the method is to use the capture/compare capability of Timer 2 in the µC to provide the timing relationship between the Data-In and Clk signals. The principal control of the interface comes from subroutine S16BIT in **Listing 1**.

To program the potentiometer, the µC must send 2 bytes to it—the "channel address," followed by the attenuation value with its most significant bit first. Sending a byte starts by loading the number of bits to send into the µC's BCOUNT register and initiating the P1 lines (setting P1.0 through P1.3 to a low state). Next, Timer 2 starts with overload enabled. The routine sets two digital compare/capture units (CC1 and

FIGURE 1



The rising edge of the clock signal validates data loading into the Data-In and Load/Shift pins of the potentiometer.

CC3 in the μC) to the "compare" mode by writing 88H into the CCEN register. The contents of the register decrease after each interrupt. In this way, eight consecutive interrupts occur, each to send 1 bit of data. The interrupt subroutine at address 006BH manages the transmission of the data bits via P1.2.

The CC1 unit generates the Clk signal on pin P1.1 when the contents of the Timer 2 count register (composed of TH2 and TL2 8-bit registers) equal the values set in registers CCL1 and CCH1. This value depends on the length of the interrupt subroutine. After transmission of the last bit, the routine stops Timer 2 by setting the value 0ECH in register T2CON. During data transmission, the main program spends its time waiting in the loop, as long as the bit FLAGS.0 is at logic 1. This bit clears in the last pass of the subroutine and sets just before Timer 2 starts. Transmission of the second byte of data occurs in exactly the same way after the routine reprograms the related registers. The Channel and Volume registers hold the 2 bytes to send.

Figures 2 and **3** show the timing relationships of the interface. In **Figure 2**, the 2 bytes Channel and Volume are 05AH and 081H, respectively. Data is valid on the rising edge of the Clk signal. **Figure 3** shows the time dependence in the interrupt-routine calls and the Clk rising edges. In this design, it takes approximately 740 μsec to program 2 bytes into the potentiometer with an 8-MHz clock frequency (a 1.5-μsec machine-cycle time). (DI #2256). EDN

To Vote For This Design, Circle No. 400

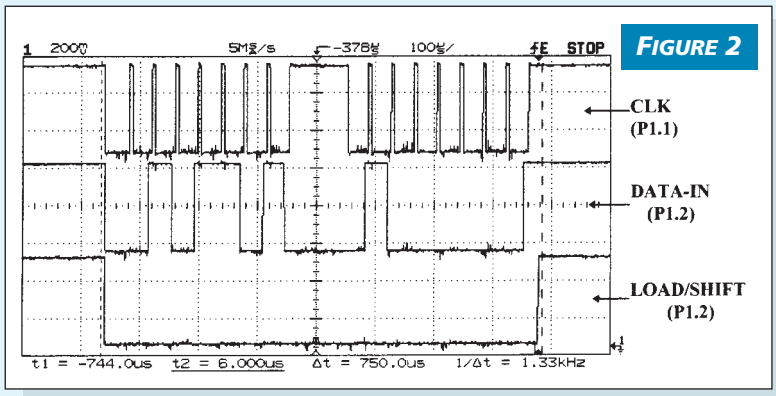


FIGURE 2

The rising edge of the clock signal appears approximately 32 μsec after the CC3 compare/capture unit in the μC generates an interrupt.

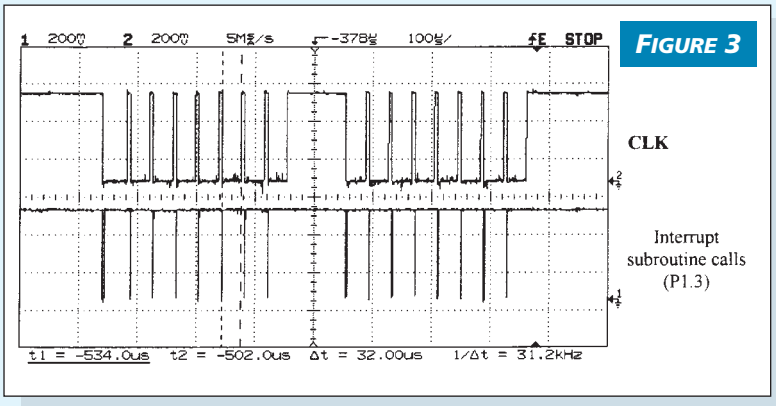


FIGURE 3

The subroutine in Listing 1 provides a simple serial digital interface to National's (and others') digital potentiometers.

LISTING 1—S16BIT SUBROUTINE

```
                ;************************REGISTERS ADDRESSES
                        DEFINITIONS************************

.EQU FLAGS,002FH
.EQU BCOUNT,004FH
.EQU CHANNEL,004DH
.EQU VOLUME,004EH

                ;*****************************INTERRUPT SUBROUTINE
                        HANDLER*****************************
;THE NUMBER OF MACHINE CYCLES TO EXECUTE THE COMMAND ARE
GIVEN AT THE ;RIGHT-HAND SIDE. THREE ADDITIONAL CYCLES ARE ADDED
FOR THE RESPONSE TO ;THE INTERRUPT REQUEST

.ORG 006BH
        CLR IEN0.7; DISABLE ALL INTERRUPTS          1
        PUSH ACC;                                   2
        PUSH PSW;                                   2
        MOV A,BCOUNT;                               1
        JZ ALL_BY; TEST IF THE LAST BIT TO SEND     2 -> 8 + 3 = 11
        DEC A;                                      1
        MOV BCOUNT,A;                               1
        MOV A,CHANNEL;                              1
        RLC A;                                      1
        MOV P1.2,C; SEND THE BIT                    2 -> 14 + 3 = 17
        MOV CHANNEL,A;                              1
        SETB IEN0.7;                                1
        POP PSW;                                    2
        POP ACC;                                    2 -> 22 + 3 = 25
        RETI;                                       2
ALL_BY: MOV A,CHANNEL; THE LAST BIT TO SEND         1
        RLC A;                                      1
        MOV P1.2,C; SEND THE BIT                    2 -> 12 + 3 = 15
        CLR FLAGS.0; INFORM S16BIT FUNCTION         1
        POP PSW;                                    2
        POP ACC;                                    2
        ANL T2CON,#0ECH; STOP TIMER 2               2
        SETB IEN0.7; ENABLE INTERRUPTS              1
        RETI;                                       2 -> 22 + 3 = 25
```

```
;*************************************************************************
;THIS COMMANDS SHOULD BE PLACED SOMEWHERE IN THE BEGINING OF
THE MAIN ;PROGRAMM
        MOV SP,#06H; SET THE STACK POINTER
        MOV IP0,#020H; SET THE INTERRUPT PRIORITY LEVEL OF CC3
        MOV FLAGS,#00H; CLEAR ALL BITS
```

```
                ;**********************LM1972 INTERFACE CONTROLLING
                        FUNCTION**********************

S16BIT: MOV TL2,#(255 - 25);SET TIMER 2 COUNT REGISTER
        MOV TH2,#0FFH
        MOV CRCL,#(255 - 25);SET TIMER 2 RELOAD REGISTER
        MOV CRCH,#0FFH
        MOV CCL1,#0FCH; SET VALUE FOR CC1 - CLK SIGNAL
        MOV CCH1,#0FFH;
        MOV CCL3,#(255 - 25); SET THE BEGINING OF INTERRUPT
        MOV CCH3,#0FFH
        MOV BCOUNT,#07H; NUMBER OF BITS TO SEND - 1
        SETB FLAGS.0
        ANL P1,#070H; INITIATE PORT 1
        MOV CCEN,#088H; ENABLE COMPARE MODE FOR CC1 AND CC
        SETB IEN1.5; ENABLE INTERRUPT FROM CC3
        ORL T2CON,#011H; START TIMER 2 WITH OVERLOAD ENABLED
S16_1:  JB FLAGS.0,S16_1; WAIT UNTIL THE FIRST BYTE IS SEND
        MOV TL2,#(255 - 25); REINITIATE TIMER 2 COUNT REGISTER
        MOV TH2,#0FFH
        MOV BCOUNT,#07H; REINITIATE NUMBER OF BITS TO SEND
        MOV A,VOLUME; SWAP VOLUME AND CHANNEL CONTENTS
        MOV CHANNEL,A
        SETB FLAGS.0
        ORL T2CON,#011H; START TIMER 2 WITH OVERLOAD ENABLED
S16_2:  JB FLAGS.0,S16_2; WAIT UNTIL THE SECOND BYTE IS SEND
        MOV CCEN,#00H; DISABLE CC1 AND CC3 UNITS
        CLR IEN1.5; DISABLE CC3 INTERRUPT
        ORL P1,#07H; SET OUTPUT PINS TO THE HIGH STATE
S_END:  RET
```

Timer inputs double as interrupt-request lines

SK SHENOY, NAVAL PHYSICAL AND OCEANOGRAPHIC LABORATORY, KOCHI, INDIA

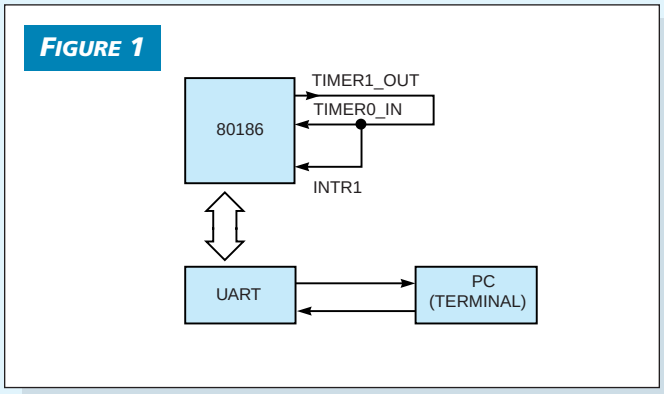
The Intel 80186 is a highly integrated 16-bit µP that is common to embedded applications. This µP's built-in interrupt controller has four interrupt-related pins that you can configure in various ways to achieve the maximum of four maskable interrupt-request lines.

In applications that require more than four interrupt-request lines, the only way out is to add an external interrupt controller, such as the 8259A. However, an alternative approach exists: You can through proper programming make the 80186's two timer-input pins function like normal edge-triggered interrupt request lines. The only penalty is an additional latency of about 1 µsec for an 8-MHz CPU, which is insignificant for many applications. This pseudo interrupt-request line is useful for interfacing any device (UART, DMA controller, coprocessor, or DSP processor) that works with internally vectored, edge-triggered interrupts.

The idea is based on the fact that you can program the timer in a mode during which the timer count resets and then the timer starts counting on a 0-to-1 transition on its input pin. Further, you can set the timer to interrupt when it reaches the value set in its Max Count register. Thus, if the Max Count setting equals 1, the timer generates an interrupt immediately after one timer-clock period from the 0-to-1 transition on the input pin. For an 8-MHz system, with the timer using the internal clock (which is one-fourth the CPU clock), the interrupt time is 500 nsec. The µP can serve the timer interrupt as any other normal interrupt would. The µP can also selectively mask and unmask the interrupt using the timer/interrupt-control registers.

You can download a demo program from *EDN*'s Web site, www.ednmag.com. (At the registered-user area, go into the Software Center to download the file from DI-SIG, #2277.) The program demonstrates the use of the Timer0_In line as a pseudo interrupt-request line. The program, along with the setup in **Figure 1** (which uses the Timer0_out line to generate the interrupt), also compares the latency of this pseudo interrupt with the latency of the normal interrupt line, Intr1. The program was written and compiled using Intel's IC86 compiler and was tested on an 8-MHz 80186 system. The same technique should work for µPs and µCs that have similar timer capabilities. (DI #2277) EDN

FIGURE 1



With the proper programming, the Timer0_In input serves as a pseudo interrupt-request line.

To Vote For This Design, Circle No. 401

DSP algorithm measures frequency and damping

OLGA BELOUSOVA, LOS ALAMOS, NM, AND ALEXANDER BELOUSOV, NEW YORK, NY

The time-domain DSP algorithm described here allows you to measure the key parameters—natural frequency and damping—in linear, second-order electromechanical systems. The method applies to a range of electromechanical transducers (electromagnetic or electrostatic), including dynamic speakers, seismic geophones, micromachined sensors, and other systems. The algorithm is based on the integral transforms of the terminal voltage of the transducers in a free transient mode (after application and removal of the step-function stimulus). It provides high immunity to both electrical noise and mechanical vibration.

Compared with traditional FFT methods, the algorithm significantly simplifies the computational task, provides better resolution in locating the spectral peak (which corresponds with the natural resonant frequency of the transducer), and allows you to calculate the damping coefficient (which you can not directly extract from an FFT). **Figure 1** uses an electromagnetic transducer, stimulated by a step-current function, with sequential integration of its terminal voltage in a free transient mode. The general equation (assuming zero initial phase) is:

$$v(t) = V_O \cdot \exp(-w_0 bt) \cdot \mathrm{SIN}(w_0(1 - b^2)^{0.5} t),$$

where V_O is the final amplitude of the terminal voltage, β is the damping coefficient (an unknown value), and ω_0 is the natural angular resonant frequency of the transducer system. The algorithm is based on the following integral transforms of the terminal voltage, v(t):

$$J_1 = \int_0^{\infty} v(t)dt,$$

$$J_2 = \int_0^{\infty} |v(t)|dt,$$

$$J_3 = \int_0^{\infty} (v(t))^2 dt.$$

Omitting the intermediate math, you can write the computational formulas for β and ω_0 as

$$b = (1 + p^2(\ln((J_2 + J_1)/(J_2 - J_1)))^{-2})^{-0.5},$$

$$w_0 = 4b \bullet J_3 / J_1^2 .$$

For a practical implementation in DSP form, you must substitute the indefinite integrals J_1 through J_3 with corresponding finite sums S_1 through S_3. The ADC must digitize the terminal voltage during a period of time long enough to allow the free transient to settle. Then, the DSP must calculate the three finite sums using the following equations:

$$S_1 = \sum (v_i), S_2 = \sum |v_i|, \text{ and } S_3 = \sum (v_i)^2.$$

The final computational formulas are as follows:

$$b = (1 + p^2(\ln((S_2 + S_1)/(S_2 - S_1)))^{-2})^{-0.5},$$

$$w_0 = (4b \bullet S_3 / S_1^2) / Dt,$$

where Δt is the sampling period.

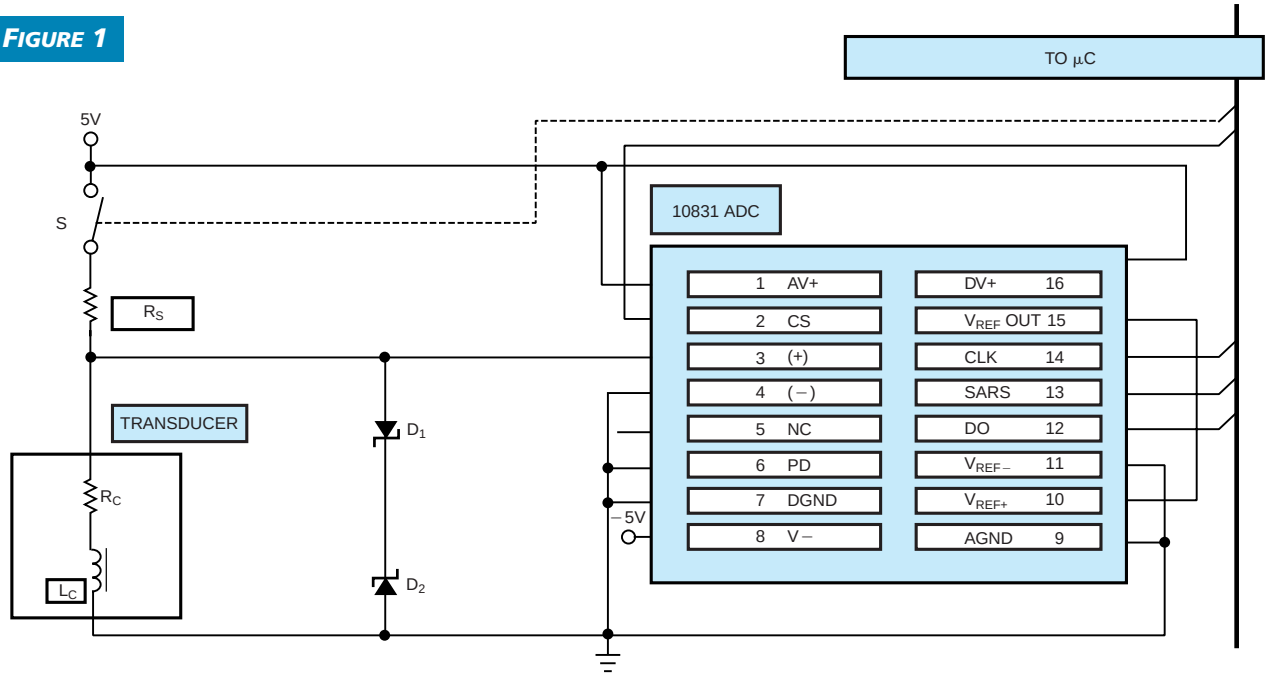
The circuit in **Figure 1** consists of the transducer under test, shown as inductor L_C in series with the coil resistance, R_C; a current-stimulus circuit (analog switch S with a current-limiting resistor, R_S, and the integrated ADC. We chose the low-power 10831 ADC from National Semiconductor (www.national.com) because of its convenient serial interface to the μC. Zener diodes D_1 and D_2 (approximately 5V breakdown) protect the ADC's input against overvoltage from inductive spikes).

The principal of operation is simple. First, you apply the current step function to the transducer. The duration of the current stimulus should be long enough to allow the transient to settle; this value depends on the estimated natural frequency and damping in the system. For example, for a typical seismic geophone with f_0=10 Hz and β=0.6, the step function should last 0.5 to 1 sec. The same rule applies to the measuring cycle in a free transient. In general, you can stop the measurement when the terminal voltage drops to less than 1 LSB in the chosen ADC. The sampling period, Δt, should be small enough to avoid methodical errors that accrue from substituting the analog integral transforms with discrete sums.

Because the resonant frequencies of mechanical systems are typically low, it is relatively easy to avoid the methodical errors. For example, the maximum sampling rate of the ADC 10831 is 74 kHz; thus, the resulting errors are low. You could also use a 10-bit 10732 ADC, a differential-input, single-supply device. For higher frequency and better resolution, you could use a 12-bit 12130 ADC, which has 14-μsec throughput time. You can use any embedded-system μC for the method. (DI #2244) EDN

To Vote For This Design, Circle No. 402

FIGURE 1



A time-domain DSP algorithm measures natural frequency and damping in electromechanical systems.

Video equalizer sharpens VCR images

WAYNE SWARD, CONSULTANT, BOUNTIFUL, UT

Video signals from a VCR lose sharpness, resulting in a flat-looking image. The effect is especially noticeable when copying from one tape to another, as you do in a video-editing system. High-quality recorders minimize this effect, but they are expensive.

The circuit in **Figure 1** sharpens picture images without introducing the shadows, ringing, and noise often observed with commercial video equalizers. The key to this improvement is the coaxial delay line, which replaces the typical LC network in other equalizers.

The circuit applies the video signal to a MAX466 quad video amplifier. The first amplifier drives a sample of the original video into a resistive summing network and a second amplifier through a 3.58-MHz trap. The trap deletes the chrominance-color signal from the original video. The second amplifier applies the remaining luminance-brightness video signal to the shorted coaxial delay line. This amplifier also applies a sample of the luminance signal to the resistive summing network. The total round-trip delay of the delay line is 65 nsec, about half the typical rise time of a VCR output for typical VHS or 8-mm VCRs.

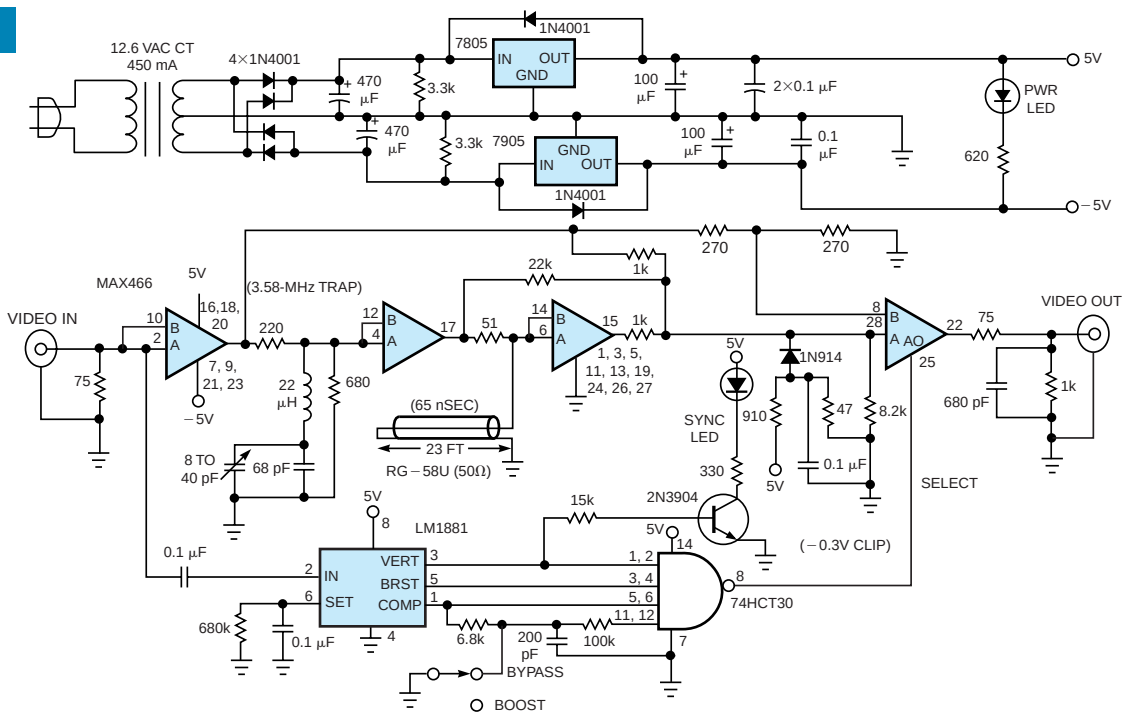
The output of this delay line drives a third video amplifier, which recovers the edges of picture images. The enhanced-edge output of this third amplifier combines some of the original video signal from the first amplifier, some of the luminance video signal from the second amplifier, and the enhanced-edge video signal from the third amplifier into an improved video signal.

The 1N914 diode and associated resistors clip the video to –0.3V, which differentiates the enhanced video from being confused with normal sync signals. The resulting video signal drives one input of the fourth video amplifier. The second input of the fourth video amplifier is the original video signal, including sync and color burst. The LM1881 video-sync detector and the 74HCT30 NAND gate-switch this fourth video amplifier so that the original sync and color burst from the first amplifier go directly to the output, and the enhanced video signal from the summing network goes to the output during active video time.

The 680-pF capacitor across the video output suppresses switching transients generated in the fourth amplifier when switching between sync/color burst and enhanced video but does not degrade the enhanced video. The circuit also contains a power LED, a vertical-sync-indicator LED, and a bypass/boost switch. The power supply is a conventional linear design. The circuit is built on a double-sided copper-clad pc board.(DI #2251) EDN

To Vote For This Design, Circle No. 403

FIGURE 1



Using a delay line in place of the typical LC equalizer network allows this circuit to sharpen VCR images without introducing the artifacts commonly produced by commercial equalizers.

Programmable charger fills capacitor banks

DENNIS FEUCHT, INNOVATIA LABS, TOWNVILLE, PA

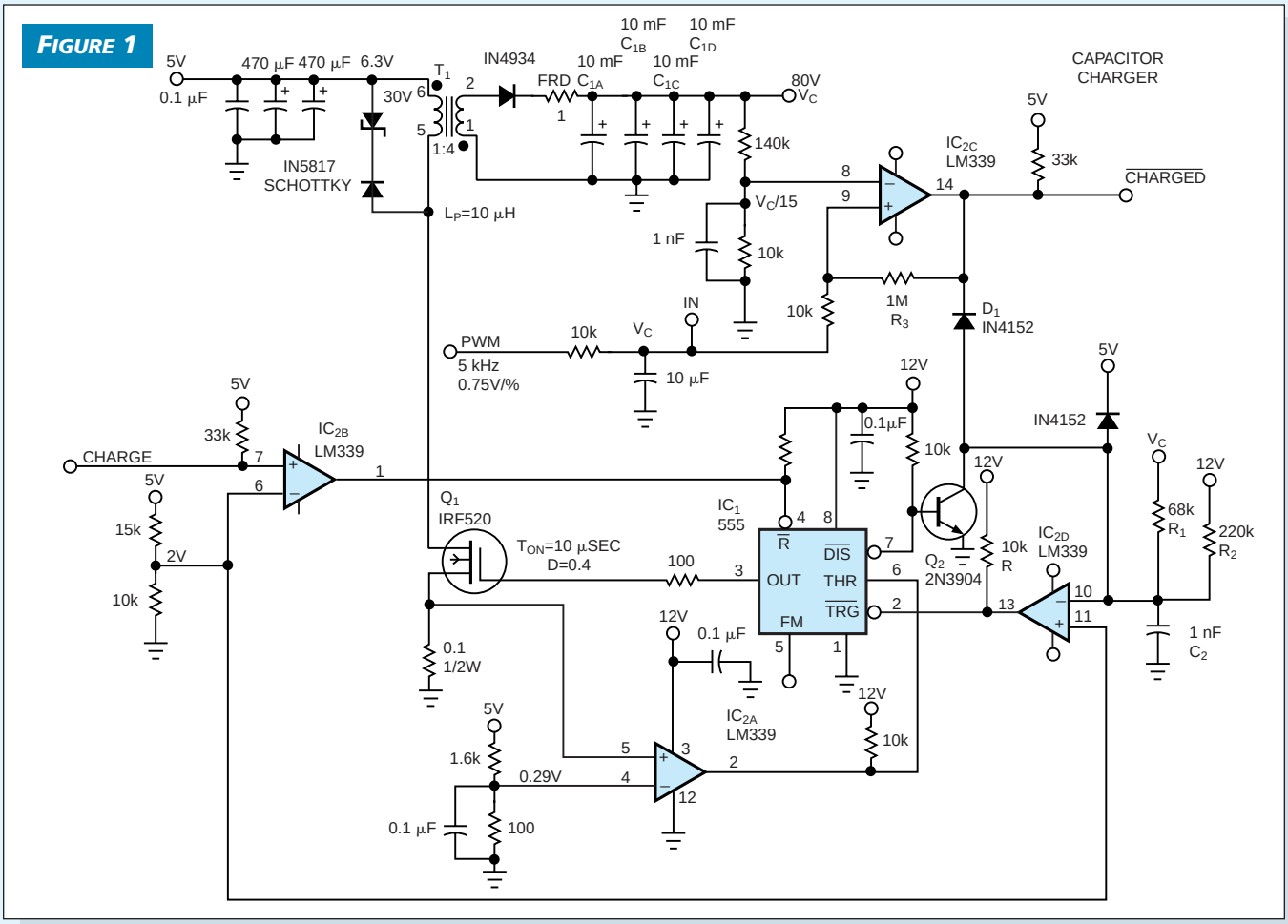
Designed for the use with a pyrotechnic pulse generator, the charger in **Figure 1** uses low-cost components to charge the C_1 capacitor bank to a voltage as high as 75V, or to 25V in a few seconds. A 555 timer, IC_1, controls a flyback converter via the Q_1 transistor switch. The converter draws its charging current from a 5V supply. The 555 turns Q_1 on until T_1's primary current reaches approximately 3A, as detected by comparator IC_{2A}, which turns IC_1 off. Then, Q_2 also turns off, and C_2 charges through R_1 and R_2, thereby setting the off-time of the charger cycle. IC_{2D} starts the new cycle by triggering IC_1.

Because R_1 connects to the charger output (V_C), as C_1 charges, the off-time decreases. The secondary conduction time decreases as V_C opposes T_1's flux. But the flyback converter transfers no charge when T_1 is not conducting current. To minimize this no-charge time, the off-time decreases with V_C. R_2 is necessary for start-up, when V_C is 0V and cannot charge C_2. A digital input (charge) turns the charger on and off.

The target voltage on C_1 depends on the duty cycle of a 5V CMOS-level pulse applied to the PWM input. A 5-kHz PWM waveform has a scale factor of 0.75V/%. A 100% duty cycle produces a 75V target voltage. IC_{2C} detects when this voltage is reached and turns the charger off through D_1. IC_{2C}'s output ($\overline{\text{Charged}}$) can also indicate the completion of charging. If you use this output as a computer input, you should monitor it in a loop, because it is unasserted during recharge. R_3 provides hysteresis, and, as C_1 leaks charge and V_C decreases, IC_{2C} changes output state and charging commences, to keep C_1 "topped off." (DI #2260). EDN

To Vote For This Design, Circle No. 404

FIGURE 1



The duty cycle of a CMOS pulse train determines the target charged voltage for a bank of high-energy capacitors.

µC provides wireless keypad control

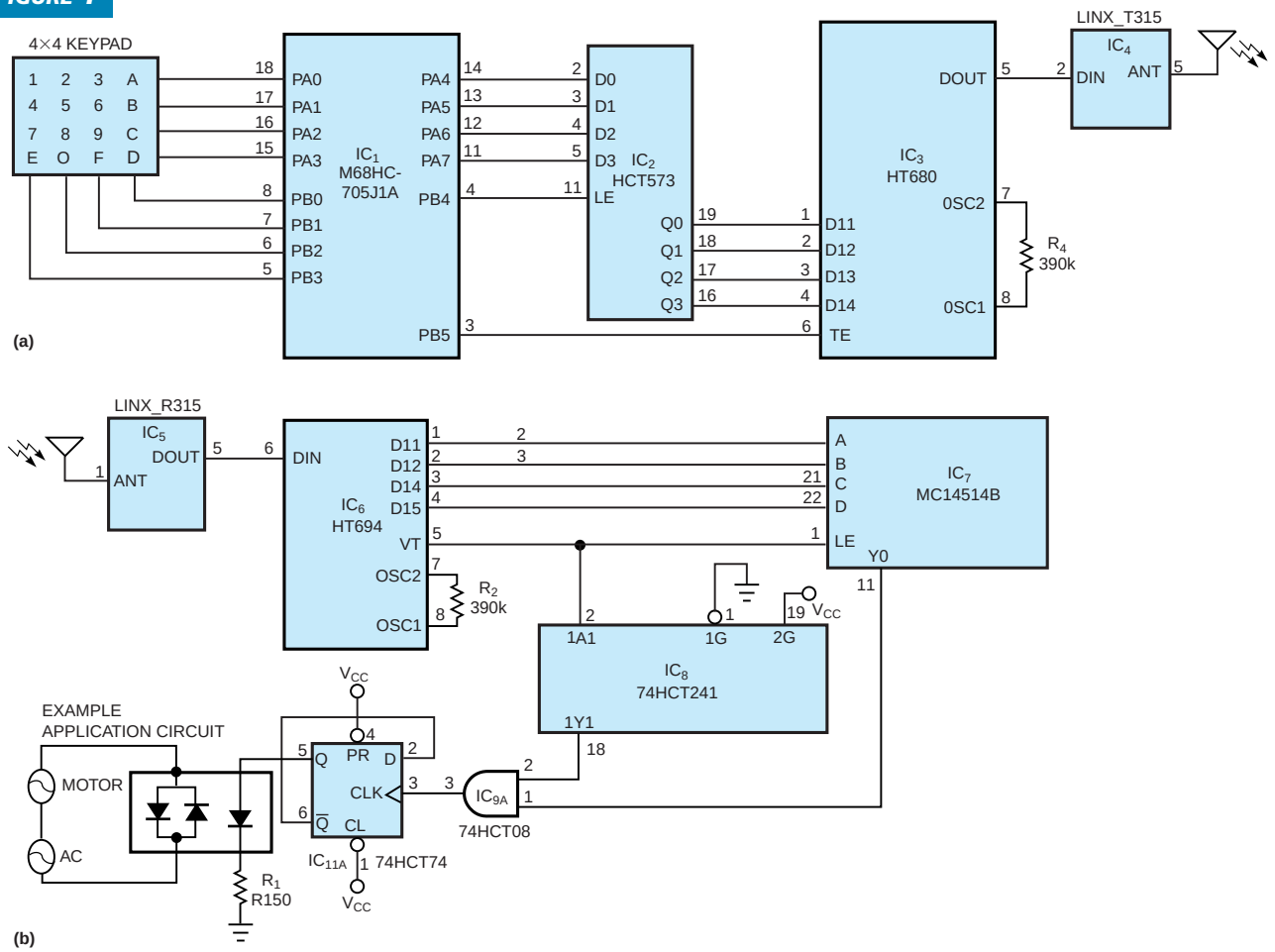
LLOYD KHUC, MOTOROLA INC, AUSTIN, TX

The circuit in **Figure 1** is a simple, 4×4 keypad remote-control system. A 68HC705J1A µC, IC_1 (**Figure 1a**), costs less than $1 and controls the keypad functions. When you depress any key, the µC provides a 4-bit hex-data output and then enables a latch signal to latch the data into IC_2. Next, the µC enables IC_2 to transmit the signal to the IC_3 encoder to convert the 4-bit hex data to serial data to send to IC_4. IC_4, the RF data-transmitter module, mixes the serial data with the 315-MHz carrier frequency to transmit.

In the receiver circuit (**Figure 1b**), the IC_5 data-receiver module removes the 315-MHz carrier frequency from the receiver signal, and IC_6 decodes the serial data into a 4-bit hex parallel data output. IC_7 converts the 4-bit hex data into 16 data bits to control 16 application circuits. IC_3 and IC_6 are a matching encoder/decoder pair that eliminates any unwanted interference frequencies. IC_4 and IC_5 are a matching RF data-transmitter/receiver pair with a 315-MHz carrier frequency. You can download the assembler code for the µC system from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the files from DI-SIG, #2261. (DI #2261). EDN

To Vote For This Design, Circle No. 405

FIGURE 1



A simple 4X4 keypad gives you wireless control of 16 lines, using an inexpensive µC, a few logic blocks, and a transmitter/receiver module pair.

EDITED BY BILL TRAVIS & ANNE WATSON SWAGER

Ethernet 10BaseT simulator jig yields zero emissions

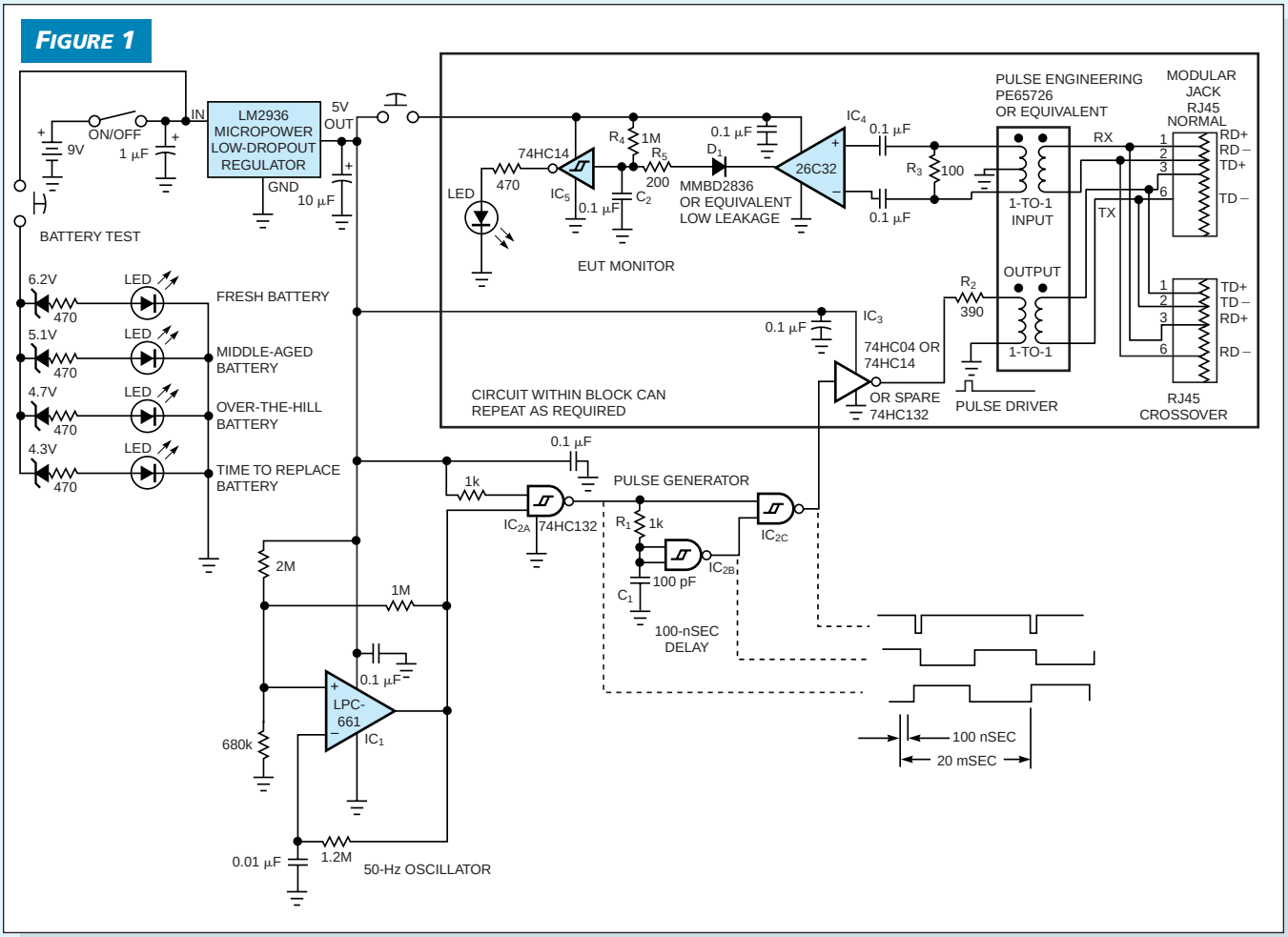
GLEN CHENIER, FUJITSU NETWORK COMMUNICATIONS, RICHARDSON, TX

A test jig **(Figure 1)** is a valuable tool because it evaluates RF emissions from Ethernet unshielded-twisted-pair (UTP) 10BaseT LAN-interface devices without contaminating the measured results with its own RF emissions. When an RF-emissions-measurement lab tests a multiport, UTP 10BaseT Ethernet device for compliance with FCC-radiated emission limits, the test is meaningful only if the device transmits data packets from all 10BaseT ports. To enable this transmission, the 10BaseT ports must receive a steady stream of link test pulses from attached 10BaseT devices. Unfortunately, the attached devices commonly radiate from their attached cables on the same frequencies as the equipment under test (EUT). This problem makes EUT performance evaluation and any trial fixes difficult, if not impossible. The solution is to eliminate the radiated noise from the ancillary equipment.

A test generator outside the measurement area can avoid interfering emissions from the generator and its cables by generating the packet data and feeding it over optical fiber to an EUT port. But the EUT 10BaseT ports still need a source of link test pulses to retransmit the data packets. To evaluate EUT emissions from the cables, these sources must connect to the EUT by UTP cables of at least 1m. Only emission-free link-test-pulse sources provide an accurate evaluation of the EUT. The emission-free requirement is difficult to meet when the link-test-pulse sources are 10BaseT devices with noisy 20-MHz clock oscillators and digital circuitry.

The circuit in **Figure 1** generates the required link-test pulses without RF emissions. The pulse must have a width of 60

FIGURE 1



A useful test jig, which includes an equipment-under-test (EUT) monitor and pulse-driver circuit, evaluates RF emissions from Ethernet unshielded-twisted-pair LAN-interface devices without contaminating the measured results with its own RF-emissions.

to 130 nsec with a repetition frequency of 42 to 125 Hz. Pulse amplitude should be 500 mV to 3V. This test jig outputs pulses of about 1V, but you can easily change this level by changing the value of R_2.

For convenience, the jig operates from a 9V battery. The use of CMOS devices results in a current drain of less than 100 μA. According to battery specifications, this current drain translates to an expected battery life of several thousand hours. The duty cycle of the link test pulses is 100 nsec/20 msec=0.0005%, so little battery energy is necessary to drive the EUT receivers. A pushbutton battery-test feature using zener diodes and a four-LED bar-graph display ensures that the battery is capable of the day's testing. To extend battery life, a pushbutton EUT-monitor circuit verifies that the EUT is working properly and is actually transmitting data packets from each port you test.

The frequency of the master oscillator (IC_1) is 50 Hz, which spaces the link-test pulses at 20-msec intervals. This low frequency has low harmonic energy in the 30-MHz and higher portion of the RF spectrum; thus, it achieves the primary design goal. IC_1, a micropower LPC661, has low power consumption, yet its slew rate is fast enough that the input of the following 74HC132 Schmitt trigger stage (IC_{2A}) is between the rails for a minimum time.

You can eliminate IC_1 and use IC_{2A} as the oscillator, but the power consumptions of HCMOS gates and Schmitt triggers tend to rise drastically when the input voltages are not at supply rails. A Schmitt-trigger CMOS oscillator has a constant sawtooth centered linearly between its trip points at its input. The measured difference in the jig's total power drain is 6.5 mA with the HCMOS oscillator and 85 μA with the LPC661 oscillator. Long battery life is a secondary design goal.

The pulse generator uses a 100-nsec RC-delay line (R_1, C_1), and a Schmitt-trigger buffer (IC_{2B}) to present the 50-Hz square wave and a 100-nsec delayed version of this square wave to the inputs of NAND gate IC_{2C}. Thus, IC_{2C}'s output consists of 100-nsec-wide pulses at 20-msec intervals. The pulse-driver gate (IC_3) inverts these pulses to drive the output transformer through a 390Ω current-limiting resistor (R_2). The transformer is a 1-to-1 Ethernet transformer. You can use other types of transformers with built-in filters; however, the low pulse-repetition rate does not require a bandlimiting filter. Changing the value of R_2 varies the nominally 1V output's amplitude.

Two RJ45 jacks connect the simulator to the EUT. The test jig includes both normal and crossover-wired jacks so that you always have the jack you need for the available UTP cable.

The EUT-monitor circuit includes the 100Ω termination (R_3) for the EUT-transmit pair. Keep the leads in this area short and symmetrical to avoid reradiating the data packets from the EUT and causing false emission readings. Even with V_{CC} off, IC_4 has a high input impedance to avoid generating and reradiating harmonics of the data packets. When you depress the EUT-monitor test button, any input data activity causes the IC_4's output to toggle at the packet data rate. The negative-going data pulses discharge C_2, resulting in a high output from IC_5's Schmitt-trigger inverter, which in turn lights the LED.

R_5, a 200Ω resistor, slows the capacitor's discharge time constant enough to ensure that any overshoot or ringing from the EUT's positive-going, individual link test pulses do not light the LED. Only actual data packets have a pulse density sufficient to light the LED. Once discharged, the low-leakage diode, D_1, and the 1M R_4 increase the capacitor's recharge time (the period after the end of the packet when IC_4's output is high again). The lengthy recharge time extends the LED's on-time to 100 msec for maximum visibility.

You can add any number of pulse-driver and EUT-monitor sections to the basic circuit, depending on the number of simulator ports and the limits of the LM2936 regulator when all LEDs are on. If you wish to power the EUT-monitor circuit from a separate higher power regulator, place the pushbutton at the additional regulator input so that the regulator's quiescent-current drain does not unnecessarily load the battery. If you don't wish to incorporate the EUT-monitor circuit, you still need to place the 100Ω termination resistor across the EUT-transmit pair, but you will no longer need the input transformer. (DI #2235) EDN

To Vote For This Design, Circle No. 347

Digital potentiometer autonulls op amp

STEPHEN WOODWARD, UNIVERSITY OF NORTH CAROLINA, CHAPEL HILL, NC

Op-amp applications that need the highest possible dc accuracy are generally best served by CMOS chopper-stabilized amplifiers, such as the LTC1050. But high-speed, low-noise applications may require high-performance rockets, such as the 700-MHz LT1226. So what to do for applications that need it *all*? Sometimes, a composite topology in which a bipolar amplifier provides gain-bandwidth and a CMOS chopper acts as an offset-nulling servo can do the job. Such arrangements can successfully null out offset-voltage errors. But these circuits can get messy if you also need bias-current-related error correction. The circuit in **Figure 1** offers an error-cancellation method that handles both error sources.

The circuit consists of op amp IC_1 (for example, Linear Technology's LT1226), CMOS multiplexer S_C (one-third of an HC4053), and digital potentiometer P_1 (Xicor's X9C103). The topology supports two modes of operation, as selected by the TTL/CMOS-compatible NADJ signal. NADJ=0 connects IC_1 as a standard noninverting gain block. The circuit values shown, combined with the impressive specs of the frequency-compensated LT1226, provide a gain of 1001 with bandwidth extending from dc to beyond 500 kHz and input-related noise of approximately 2 $nV\sqrt{Hz}$.

Null-adjustment mode occurs when NADJ=1 disconnects the input source and effectively causes IC_1's output to run open-loop. IC_1's output and then slews to one rail or the other, as determined by the sign of its net offset error. If $R_3=R_S-R_1||R_2$, where R_S is the dc source resistance, then IC_1's output reflects the sum of both voltage *and* current bias errors. The circuit level-shifts and filters IC_1's output and applies it to the up/down control input of P_1. This action sets up P_1's internal up/down-counter logic to increment or decrement one step, depending on the state of IC_1's output and thus on the sign of IC_1's offset. The counter step occurs on the subsequent NADJ=0 transition.

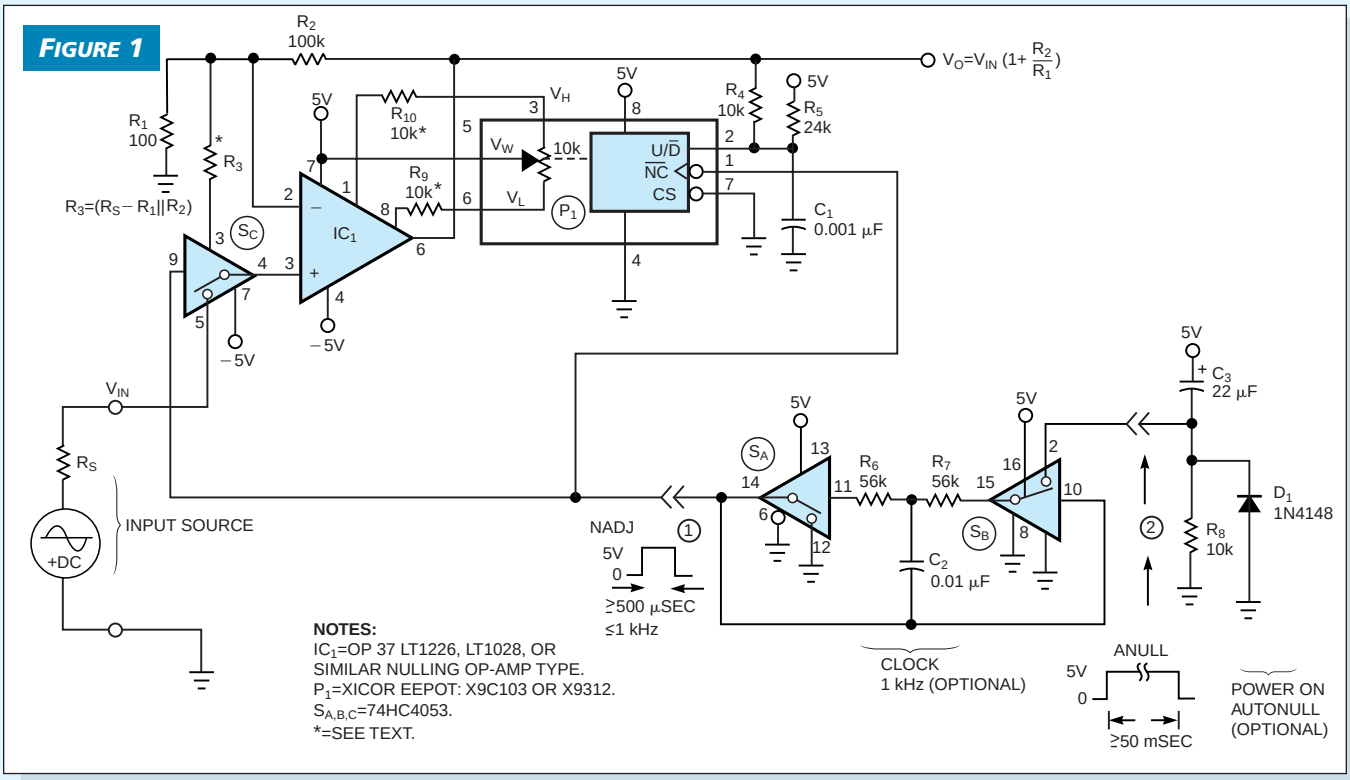
The connection of the V_L,V_W, and V_H terminals of P_1 to the nulling terminals of IC_1 closes a feedback loop that tends to push IC_1 one step toward null for every I/O cycle of NADJ. Because the X9C103 has 100 resolved settings, the technique requires a maximum of 99 NADJ pulses to complete the nulling process. After nulling, P_1 retains the final null setting in digital memory as long as the 5V supply remains connected or until the nulling process repeats. Observed performance reveals that using an OP37 consistently achieves residual-offset errors of less than 5 µV.

If it is inconvenient to provide an external NADJ clock source in a given application, you can add the S_A/S_B multivibrator at Node 1. This 1-kHz clock circuit receives its gating from the CMOS-compatible anull signal, such that anull=1 enables continuous null adjustment, and anull=0 enables normal amplifier operation. The maximum anull duration required to achieve initial null is 100 msec. If desired, you can also include D_1, R_8, and C_3 at Node 2 to provide an automatic null on each power-up cycle.

Although **Figure 1** shows an LT1226, the circuit works without modification with an OP37 and an LT1028. The circuit is also pin-compatible with the popular LT1128, OP07, OP77, OP177, and µ725 op amps. With these op amps, however, the circuit may require a slower NADJ clock rate and a longer nulling interval (increase C_2 and C_3), because of the lower gain-bandwidth product of these compensated types. The circuit can accommodate many other op-amp types with a simple change of pin connections. The circuit can handle 15V positive-rail operations by substituting an X9312 for the X9C103 with no other changes. (DI #2262). EDN

To Vote For This Design, Circle No. 348

FIGURE 1



Have your cake (high speed) and eat it (low dc errors) too, with this autonulling circuit, using a digitally controlled potentiometer.

Single wire connects microcontrollers

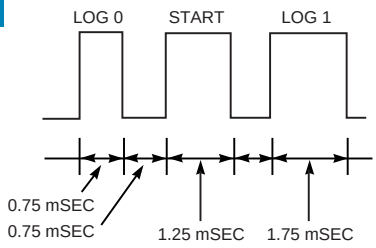
ABEL RAYNUS, ARMITRON INTERNATIONAL, MELROSE, MA

Low-cost µCs, such as Motorola's 68HC705 Series, offer great simplicity at the expense of some useful functions—notably, serial data transmission. Unlike their predecessors, these µCs do not have serial communication interfaces (SCIs), serial peripheral interfaces (SPIs), or simple serial I/O ports (SIOPs). This method describes how you can overcome this deficiency by creating an asynchronous serial interface through µC software. The most obvious way to effect the interface is to use pulse-width coding to differentiate the start pulse and the logic 1 and 0 pulses. You can use any value of pulse-width ratio, depending on your design objectives. This application uses the 1-to-2-to-3 ratio, slightly modified for easy programming. So, logic 0, start, and logic 1 have widths of 0.75, 1.25, and 1.75 msec, respectively (**Figure 1**).

This example uses 68HC705J1A µCs. The information to transmit accumulates in the output data register of the transmitting µC (**Figure 2**). The data-transmission subroutine (**Listing 1** shows a fragment) generates the start pulse, which is followed by an 8-bit data word that reflects the state of the output data register. This pulse sequence goes, LSB-first, to the data-in input of the receiving µC. In the µC transmitter, you can use any output pin as data out. In the µC receiver, you can use any one of the four lower PortA pins (PA0 through PA3) as data in.

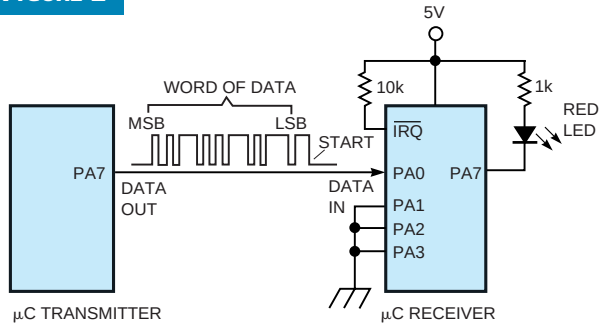
You should program the input pins as positive-edge, external-interrupt inputs. Because pins PA0 to PA3 combine in a logic-OR operation in the µC, you should connect the unused pins to ground to avoid false interruption. You should disable the $\overline{IRQ}$ pin by connecting it to 5V. The external-interrupt subroutine (**Listing 2**) restores the data word, which can generate the proper response according to your design objectives. **Listing 3** shows a fragment of the receiver routine. The program's watchdog utility lights a red LED to indicate that the communication link between the µCs is broken or that it received the wrong sequence. The method also applies to

FIGURE 1



Pulse-width ratios provide a convenient way to transmit serial data between µCs lacking communications amenities.

FIGURE 2



With the help of some µC software, a simple one-wire connection provides serial communications between low-cost µCs.

LISTING 1—TRANSMITTER-PROGRAM FRAGMENT

```
                    1  ***********************************
                    2  *  TRANSMITTER PROGRAM FRAGMENT   *
                    3  ***********************************
                    4  * Generates the sequence of start pulse and 8-bit word
                    5  * of data according to the content of register REG.
                    6  ***************************************************
                    7  *nolist
0000                8  $include "std-j1a.asm"
                    9  *list
                   10  *  I/O PORTS
0000               11  data_out  equ  7  ;prtA
                   12  *  VARIABLES
00C0               13          org     RAM
00C0               14  REG     rmb     1 ;output data register
00C1               15  num     rmb     1 ;bit test register
                   16  *  INITIALIZATION
07F1               17          org  MOR
07F1 00            18          fcb  $00
0300               19          org  ROM
0300 9C            20  init    rsp     ;reset stack pointer to $ff
0301 A680          21          lda     #$80 ; pA7 as output
0303 B704          22          sta     ddrA
0305 3F00          23          clr     prtA
0307 3FC0          24          clr     REG
0309 3FC1          25          clr     num
                   26  *  DATA TRANSMISSION SUBROUTINE
030B AE7D          27          ldx     #125T   ;start pulse (1.25ms)
030D CD0335        28          jsr     pulse
0310 A601          29          lda     #$01    ;0-bit test prepare
0312 B7C1          30          sta     num
0314 B6C0          31  w1      lda     REG     ;is tested bit = 0?
0316 B4C1          32          and     num
0318 270B          33          beq     w2
031A AEAF          34          ldx     #175T   ;logic1 pulse (1.75ms)
031C CD0335        35          jsr     pulse
031F 98            36  w3      clc             ;0 -> C-carry bit
0320 38C1          37          lsl     num     ;go to next tested bit
0322 24F0          38          bcc     w1      ;is it NOT a last bit?
0324 81            39          rts     ;return from DATA TRANSMISSION
0325 AE4B          40  w2      ldx     #75T    ;log0 pulse (0.75ms)
0327 CD0335        41          jsr     pulse
032A 20F3          42          bra     w3
                   43  ***********************************************
032C A602          44  dly01x  lda     #2      ;Delay =0.01*x [ms]
032E 4A            45  rep0    decA
032F 26FD          46          bne     rep0
0331 5A            47          decx
0332 26F8          48          bne     dly01x
0334 81            49          rts             ;return from dly01x
                   50  ***********************************************
0335 1E00          51  pulse   bset    data_out,prtA     ; width = x
0337 CD032C        52          jsr     dly01x
033A 1F00          53          bclr    data_out,prtA
033C AE40          54          ldx     #64T
033E CD032C        55          jsr     dly01x
0341 81            56          rts             ;return from pulse
                   57  ***********************************************
0342 80            58  un      rti     ;return from unused interrupts
07F8               59          org     VECTORS
07F8 0342          60          fdb     un      ;Timer Interrupt unused
07FA 0342          61          fdb     un      ;External Interrupt unused
07FC 0342          62          fdb     un      ;SWI unused
07FE 0300          63          fdb     init    ;set restart address
```

wireless applications with minor modifications. You can download the complete **listings** from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the files from DI-SIG, #2265. (DI #2265). EDN

To Vote For This Design, Circle No. 349

LISTING 2—EXTERNAL-INTERRUPT SUBROUTINE

```
ART10.ASM              Assembled with IASM   04/09/1998  11:26  PAGE 2



                  57  * EXTERNAL INTERRUPT SUBROUTINE
 032E 3FC1        58  ExtInt: clr     T
 0330 3FC0        59          clr     W
 0332 000011      60  e0      brset   data,prtA,e1 ;High level ?
 0335 01C31C      61          brclr   WF,flag,e2   ;WF=0?
 0338 B6C0        62          lda     W
 033A A101        63          cmp     #1           ;W=1?
 033C 2724        64          beq     log0
 033E A103        65          cmp     #3           ;W=3?
 0340 2728        66          beq     log1
 0342 11C3        67          bclr    WF,flag      ;0 -> WF
 0344 2041        68          bra     e3
 0346 3CC1        69  e1      inc     T
 0348 B6C1        70          lda     T
 034A A137        71          cmp     #55T
 034C 26E4        72          bne     e0
 034E 3FC1        73          clr     T
 0350 3CC0        74          inc     W
 0352 20DE        75          bra     e0
 0354 B6C0        76  e2      lda     W
 0356 A102        77          cmp     #2           ;W=2?
 0358 262D        78          bne     e3
 035A 10C3        79          bset    WF,flag      ;1 -> WF
 035C A6FE        80          lda     #$fe         ;0 -> 0-bit of num
 035E B7C4        81          sta     num
 0360 2025        82          bra     e3
 0362 B6C5        83  log0    lda     reg          ;put 0 into given bit of
 0364 B4C4        84          and     num          ;reg. without changing
 0366 B7C5        85          sta     reg          ;of the rest of its bits.
 0368 200C        86          bra     e4
 036A B6C4        87  log1    lda     num          ;put 1 into given bit of
 036C B7C6        88          sta     mem          ;reg. without changing
 036E B4C5        89          and     reg          ;of the rest of its bits
 0370 33C6        90          com     mem
 0372 B8C6        91          eor     mem
 0374 B7C5        92          sta     reg
 0376 99          93  e4      sec                  ;1 -> Carry bit
 0377 39C4        94          rol     num          ;go to the next bit
 0379 250C        95          bcs     e3           ;is it NOT the last bit?
 037B 11C3        96          bclr    WF,flag      ;0 -> WF word process flag
 037D 17C3        97          bclr    WDF,flag
;0+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

LISTING 3—RECEIVER-PROGRAM FRAGMENT

```
                   1
                   2  *     RECEIVER  PROGRAM FRAGMENT
                   3  ********************************
                   4  * Transfers the received serial data
                   5  * into content of the register WORD.
                   6  ****************************************************
                   7  *nolist
0000               8  $include "std-jla.asm"
                   9  *list
07F1              10          org  MOR  ;pos.edge Ext.Interrupt
07F1 24           11          fcb  $24  ;  on pA0 - pA3 enable
                  12  *I/O PORTS
07F2              13  data    equ  0  ;prtA Data Input pin
07F2              14  RedLED equ  7  ;prtA red LED output pin
                  15  *Specific equates
07F2              16  WF      equ  0  ;Word processing flag
07F2              17  F1      equ  1  ;signal presense flag
07F2              18  WDF     equ  3  ;watch-dog flag
                  19  * VARIABLES
00C0              20          org      RAM
00C0              21  W       rmb      1 ;pulse width counter
00C1              22  T       rmb      1 ;time (0.5 ms) counter
00C2              23  wdc     rmb      1 ;watch-dog counter
00C3              24  flag    rmb      1 ;flag register
00C4              25  num     rmb      1 ;register to form word
00C5              26  reg     rmb      1 ;temporary word register.
00C6              27  mem     rmb      1 ;memory register
00C7              28  WORD    rmb      1 ;final received word register.
                  29  * INITIALIZATION
0300              30          org      ROM
0300 9C           31  init    rsp              ;reset stack pointer to $ff
0301 A6F0         32          lda      #$f0        ;pA0 - pA3 as input
0303 B704         33          sta      ddrA        ;pA4 - pA7 as output
0305 CD0318       34          jsr      in_set      ;go to initial set
0308 1E0A         35          bset     IRQE,ISCR   ;ExtInt enable
030A 1A08         36          bset     TOIE,TSCR   ;TOF interrupt enable
030C 9A           37          cli                  ;interrupt enable
030D 06C303       38  m0      brset    WDF,flag,m1 ;WDF=1? No data-in?
0310 00C3FA       39          brset    WF,flag,m0  ;WF=1? wait for word end
0313 CD0318       40  m1      jsr      in_set
0316 20F5         41          bra      m0
                  42  ****************************************************
0318 3F00         43  in_set  clr      prtA        ;set red LED on
031A 5F           44          clrx                 ;start to clear
031B 6FC0         45  a0      clr      RAM,x       ; 8 variables in RAM
031D 5C           46          incx
031E A308         47          cpx      #8T
0320 25F9         48          blo      a0
0322 81           49          rts                  ;return from in_set
                  50  ****************************************************
0323 3CC2         51  TOFint  inc      wdc
0325 3DC2         52          tst      wdc         ;wdc / 0 ?
0327 2602         53          bne      t0
0329 16C3         54          bset     WDF,flag    ;1 -> WDF
032B 1608         55  t0      bset     TOFR,TSCR   ;TOF reset
032D 80           56          rti                  ;return from TOFint
```

Photo-flash charger minimizes parts count

STEVEN CHENETZ, MICREL SEMICONDUCTOR, SAN JOSE, CA

Photo-flash and strobe devices operate by discharging a high-voltage capacitor into a bulb. Charging the capacitor from a battery or other low-voltage source requires a step-up dc/dc converter to boost the voltage, typically to 300V. One way to generate the high voltage is to use a flyback converter. The circuit in **Figure 1** provides a simple and reliable way to charge a high-voltage capacitor. The flyback converter performs two functions: It boosts the low-voltage input and provides isolation between the input (battery) and output (high voltage). Its main components are the power transformer; the output diode; the output capacitor; and the MIC3172 controller chip, which combines the switching transistor, voltage regulator, and control logic.

The transformer stores energy when the internal transistor of the MIC3172 turns on, allowing current to flow through the transformer's primary. When the transistor turns off, the stored energy flows through the output rectifying diode and into the capacitor. The voltage across the capacitor increases with each switching cycle until it reaches the preset voltage. The resistive divider $R_2/R_2/R_3$ and the 1.24V reference in the IC determine the preset output voltage: $V_{OUT}= V_{REF}(R_1+R_2+R_3)/R_3$.

Once the capacitor voltage reaches the preset value, the MIC3172 stops switching. Current flow in the output components cause the capacitor to discharge. The MIC3172 provides occasional energy pulses that keep the capacitor fully charged. When the capacitor discharges into the bulb, the charging process repeats. D_1 and D_2 clamp any voltage spikes

on the collector of the MIC3172 switch node, caused by leakage inductance on the transformer. When the IC's internal transistor turns off, the voltage across the transformer's primary approximately equals the output voltage divided by the turns ratio. The voltage at the transistor collector node (Pin 7) equals the reflected voltage plus the input voltage, plus the voltage spike caused by the leakage energy in the transformer: $V_{SW}=(V_{OUT}/N)+V_{IN}+V_{LEAKAGE}$.

The collector-node voltage must always be less than 65V. The zener-diode voltage is set greater than the maximum reflected voltage at the transformer primary. For **Figure 1**, the reflected voltage is 10V. The zener diode is a 12V device, approximately 20% greater than the reflected voltage. The maximum reverse voltage across D_2 equals the maximum input voltage. This diode must be an ultrafast or Schottky device, to prevent excessive losses in the diode.

The energy stored in the capacitor is $0.5CV^2$. The output power that the flyback converter requires to charge the capacitor in a period T is $(0.5CV^2)/T$. The following formula gives the approximate charging time for the converter circuit:

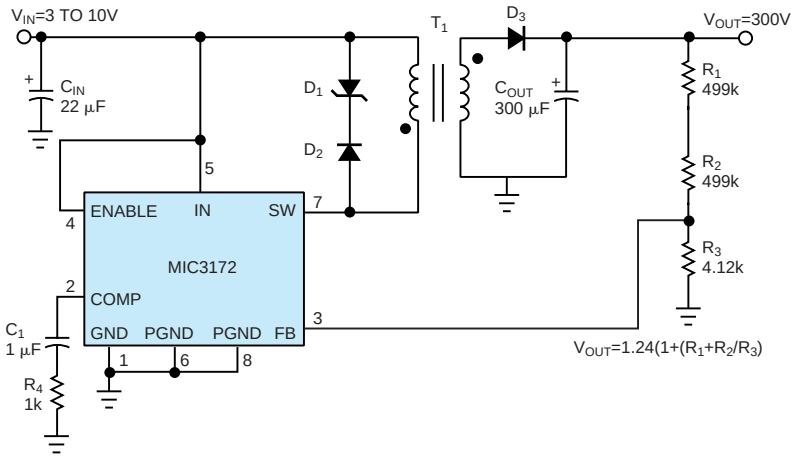
$$T_{CHARGE} = \frac{C_{OUT}\,V_{OUT}^2}{2V_{IN}I_{PK}h},$$

where I_{PEAK} is the peak current level of the MIC3172 control chip (typically, 1.8A); D is the maximum duty cycle (approximately 0.6); and η is the efficiency of the flyback converter (0.5).

Charging a 300-µF capacitor to 300V from a 5V input requires $(300\,\mu F\times 300V^2)/(2\times 5V\times 0.6\times 0.5)=5$ sec. For the circuit in **Figure 1**, the output voltage is potentially lethal. At 300V, the energy in the output capacitor is 27J, more than enough to ruin an otherwise good day. When you lay out the circuit, be sure to provide adequate spacing between the high- and low-voltage sections. The power transformer, such as the Coiltronics CTX04-13770, must have the proper spacing and insulation between the high-voltage secondary and low-voltage primary.

The circuit uses two resistors, R_1 and R_2, in the upper section of the output to reduce voltage stress, because most commonly available resistors are rated at 200 to 300V—too close to the limit for reliable, long-term operation. If R_1 or R_2 should open or if R_3 shorts, the converter runs open-loop at its maximum duty cycle. This failure mode boosts the voltage far above the preset limit and causes the output capacitor to vent. The circuit in **Figure 2** provides overvoltage protection.

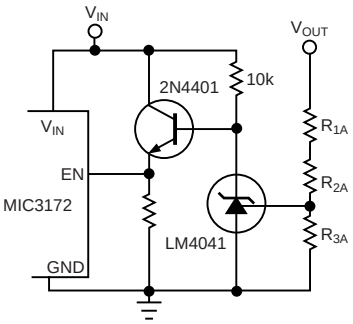
FIGURE 1



NOTES: IC=MICREL MIC3172BM BOOST-CONVERTER IC.
C_{IN}=AVX TPS SERIES TANTALUM.
T_1=1-TO-30 STEP-UP FLYBACK TRANSFORMER; LPRI=35 µH; COILTRONICS P/N CTX04-13770.
D_1=MOTOROLA 1SMB5927BT3, 12V ZENER.
D_2=MOTOROLA MBRS130LT3, 1A, 30V.
D_3=MOTOROLA MURS160T3, 1A, 600V.
R_1 TO R_4=METAL-FILM RESISTORS, 1206 (200V RATED).
C_1=CERAMIC CAPACITOR, X7R SERIES.

Generate 300V from a low-voltage source, using this simple, low-parts-count circuit.

FIGURE 2



Avoid the exploding-capacitor syndrome by using this overvoltage-protection circuit with the circuit of Figure 1.

Be sure that the resistor divider for the overvoltage circuit is separate from the voltage-regulation divider. Set the overvoltage level 15% higher than the output-voltage setting, and make sure it does not exceed the capacitor's voltage rating. (DI #2266). EDN

To Vote For This Design, Circle No. 350

Generate an analog signal with a µC

THOMAS SCHMIDT, MICROCHIP TECHNOLOGY, CHANDLER, AZ

Applications requiring D/A conversion abound, including dual-tone generation, motor-speed control, and offset-voltage generation for a sensor or for battery charging. Most designers believe the D/A converter must be either an integrated module in a µC or an external component; however, a simpler approach is possible. You can generate an analog signal by using a low-cost µC, thereby eliminating the need for external components and thus reducing board space and overall system cost. The RC network in **Figure 1** provides an easy way to convert a digital signal into an analog voltage. The RC network, a lowpass filter, connects to an I/O pin of the µC.

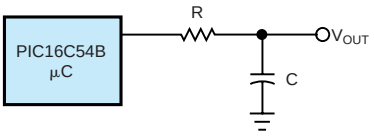
To generate an analog signal, the µC charges the capacitor via the resistor. The µC uses PWM to charge the capacitor. The voltage across the capacitor is the analog voltage. When the PWM signal is high, the capacitor charges. When the PWM signal is low, the capacitor discharges. By varying the duty cycle, you can generate a sine wave or any other analog signal. The µC in **Figure 1** is a low-cost, 8-bit RISC controller. The PWM signal, generated in software, drives the RC lowpass filter connected to one of the µC's I/O lines.

Listing 1 gives the software code. The PWM routine requires only three general-purpose registers. One register contains the value of the period; the other contains the duty cycle. The program starts by initializing the PWM output pin and the period and duty-cycle registers. The initial duty cycle is 50%. It's assumed in the initialization routine that the RC network connects to pin RA1 of the µC. After initialization, the main subroutine calls the routine in which the PWM signal is generated. In this example, the main routine calls only the PWM_Signal routine. You could easily implement other functions—a keypad or a seven-segment display, for example—just by adding call instructions for subroutines.

The PWM implementation requires a software counter. The register counter stores the software counter. The counter increments each time the program calls the PWM_Signal routine. Each time the counter increments, the program checks to see if the register's value is greater than or equal to the duty cycle. If this condition is true, the program sets the value at the port pin to logic 0. This action signifies that the time for the duty cycle has elapsed. After this time elapses, the routine checks to see if the time for the period is over. If the value of the counter is less than the value of the duty cycle, the PWM signal remains high.

If the value of the counter is greater than the value of the duty cycle, the program compares the counter with the value of the period register. If the value of the counter equals the value of the period register, the period for the PWM signal is over, and the next period starts. The performance of this PWM implementation depends on the number of times the program calls the PWM function from the main routine—the more calls, the higher the resolution. To generate a sine wave, for example, you can store the values for the duty cycle in a look-up table. The values in the look-up table depend on the values and tolerances of the resistor and capacitor and on the desired resolution of the sine wave. You can download the **listing** from *EDN*'s Web site www.ednmag.com. At the registered-user area, go into the Software Center to download the files from DI-SIG, #2268. (DI #2268). EDN

FIGURE 1



A simple RC network connected to a low-cost µC's output can generate analog signals of any desired resolution.

LISTING 1—ROUTINE FOR ANALOG-SIGNAL GENERATION

```
                        list p=pic16c54B, r=hex

                        #include <p16c5x.inc>
                        #define PWM_PORT    PORTA
                        #define PWM_PIN     0

DutyCycle       EQU     0x09
Counter         EQU     0x0A
Period          EQU     0x0B

                        ORG     0x00
Begin                   call    Initialize_PWM
Main                    call    PWM_Signal
                        goto    Main

Initialize_PWM          clrf    PORTA           ; reset PORTA
                        movlw   b'000011110'
                        tris    PORTA           ; Set PWM pin for output
movlw   80
                        movwf   DutyCycle       ; initialize duty cycle
                        movlw   FF
                        movwf   Period          ; initialize period register
                        clrf    Counter         ; reset counter
                        bsf     PORTA, PWM_PIN  ; set PWM signal to high
                        retlw   0

PWM_Signal              incf    Counter,f       ; Increment the counter
                        movf    DutyCycle,w
                        subwf   Counter,w       ; compare duty cycle against
                                                ; period
                        btfss   STATUS,C        ; Is period < duty cycle
                        retlw   0               ; duty cycle is greater than
                                                ;period, therefore PWM signal
                                                ; remains high

                        bcf     PORTA,PWM_PIN           ; time of duty cycle
                                                        ;elapsed
                        movf    Period,w                ; compare the counter
                                                        ;against the period
                        subwf   Counter,w
                        btfss   STATUS,Z                ; if the counter ==
                                                        ; period,
                        retlw   0                       ; period is not over
                        clrf    Counter                 ; reset the counter
                        bsf     PORTA, PWM_PIN          ; Set PWM_Pin to high
                        retlw   0

                        ORG     0x1FF
Reset                   goto    Begin

                        END
```

To Vote For This Design, Circle No. 351

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____ Fax _____

E-mail _____

My e-mail address may be published Yes____ No____

Company _____

_____

Address _____

_____

Country _____ ZIP _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

CIRCLE NO. 9

Edited by Bill Travis and Anne Watson Swager

Low-voltage interface circuits translate 1.8 to 5V

CC Poon and Edward Chui, Motorola SPS, Hong Kong

Interfacing two systems that operate at two arbitrary voltages is a challenging problem; there is no guarantee that one side operates at a voltage higher than the other side. Usually, the interface is an open-collector or open-drain type with just two transistors connecting back to back (**Figure 1a**). V_X is the lower of the two operating voltages. If you know which side has the lower operating voltage, the interface design is straightforward. If either side can have the lower operating voltage, you have to extract the lower one. Without the use of an op amp, you can use a diode-based circuit (**Figure 1b**). The 1N4148 is good for most applications. If a higher current capability is necessary, you can use the 1N4001. If the lower operating voltage is around 1V, D_3 should be a Schottky diode, such as the 1N5817 or MMBD701, and D_1 and D_2, can be normal PN-junction diodes.

If level translation is necessary in one direction, you can use half of the circuit for open-drain translation, which is equivalent to simple TTL. This simple circuit is fast (**Figure 2a**). When driving one standard load on a real pc board, which has approximately 10 to 20 pF of total load capacitance, the rise and fall times are fast when you view it with a scope. Circuit performance is better than that of the traditional bipolar inverter, which needs a compensation capacitor to assist turn-off. (Replacing the bipolar transistor with an enhancement MOSFET can eliminate the capacitor but results in long rise and fall times and a longer delay.) The TTL-like circuit uses only pullup resistors, which may further save pc-board space because you can use multiple pullup resistors in one resistor pack.

For a logic high-to-low transition, the delay is just the turn-on time of the transistor. For a low-to-high transition, RC effects don't appear until the output rises to about 0.5V below the lower supply voltage, V_X, when translating up (**Figure 2b**). Before that, the output tracks the input with only a $V_{CE(SAT)}$ drop, which is analogous to a cascode amplifier. The effect of turning off a saturated transistor does not manifest itself except when translating from below 1V to 5V.

The TTL-like circuit in **Figure 2a** also works well for translating from high to low voltages ($V_B<V_A$). For a high-to-low transition, the delay is just the turn-on time of the transistor. You can replace the pullup resistor by an active transistor to increase driving strength (**Figure 2c**). You must pay attention to $V_{EBO(BR)}$, which must not ex-

An open-collector or open-drain circuit (a) typically interfaces between two systems that operate at two arbitrary voltages. A diode-based circuit (b) can extract the lower operating voltage.

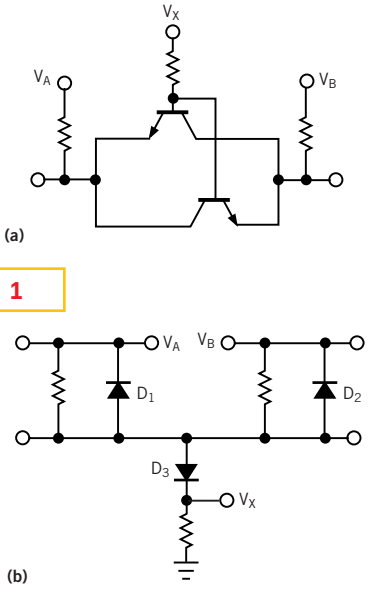


Figure 1

Level translation in one direction requires half of the open-drain translation circuit (a). For a low-to-high transition, RC effects do not appear until the output rises to about 0.5V below the lower supply voltage (b). The circuit also works for translating from high to low voltages ($V_B<V_A$), and you can replace the resistor with an active pullup (c).

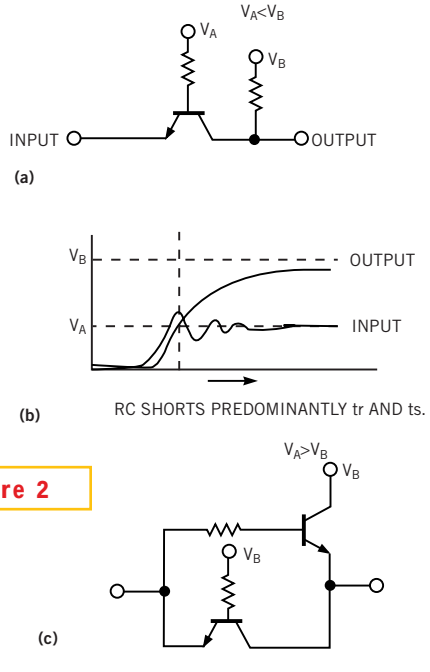
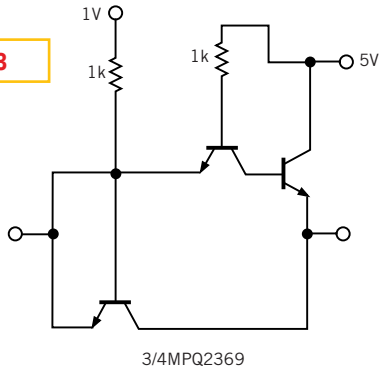


Figure 2

ceed the rated value because a violation results in premature failure of the transistor. For most small signal transistors, $V_{EBO(BR)}$ is typically 4 to 5V. Therefore, you should take care when down-shifting from 12 to 5V, such as between CMOS analog circuits and 5V logic.

The switching transistor can be MPS2369A to MPS3646 for high-speed switching. You can use the 2N3904 or BC547 for low-power applications. A 2N5458 can replace the pullup resistor at the collector if active pullup is necessary.

Figure 3



The best 1-to-5V shifting driver in the laboratory produces a typical symmetric delay of 6 nsec using three-fourths of an MPQ2369 when driving a 74AC541 buffer (**Figure 3**). (DI #2290)

To Vote For This Design,
Circle No. 500

The ultimate 1-to-5V up-shifting driver in the laboratory produces a symmetrical delay of 6 nsec using three-fourths of an MPQ2369 when driving a single 74AC541 buffer.

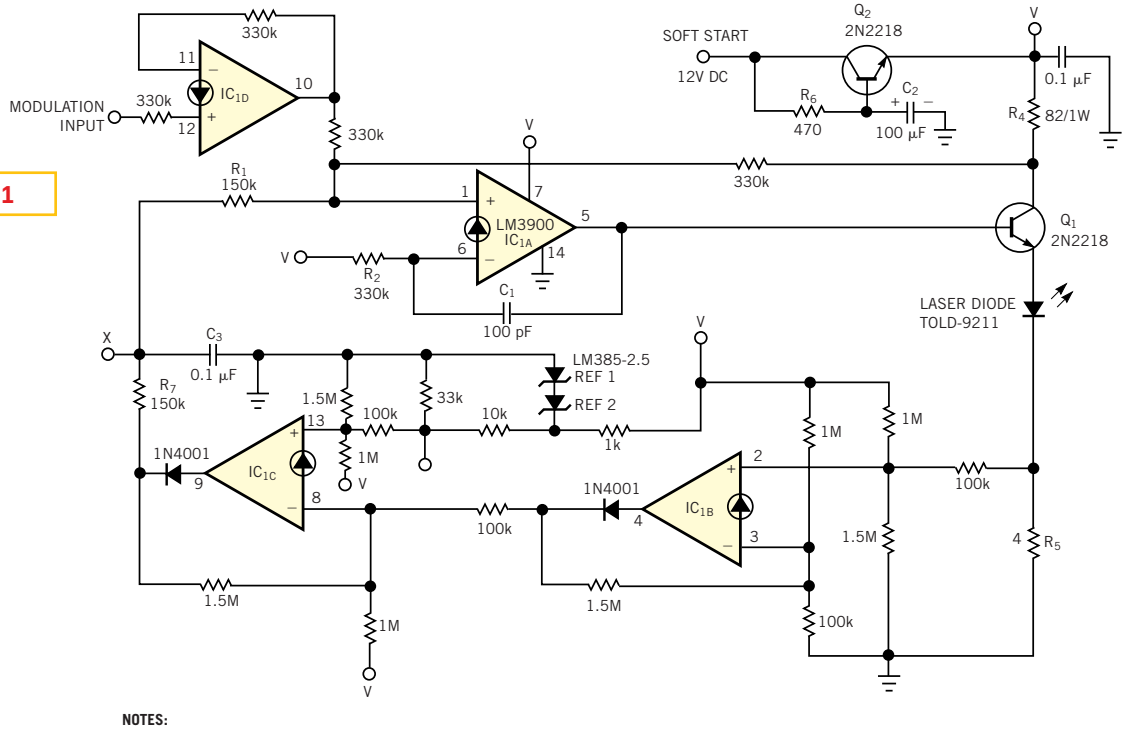
Laser-diode driver stabilizes sensitivity parameters

Anil Kumar Maini and Nita Sen, Defence Science Centre, Delhi, India

The conventional advantages of lasers—coherence, monochromaticity, and extreme compactness—make laser diodes popular in most of their potential applications. Biomedical diagnostics and high-resolution-spectroscopy applications exploit laser diodes' wavelength tunability. These applications use the laser wavelength's sensitivity to drive current and operating temperature—0.025 nm/mA and 0.3 to 0.4 nm/°C for diodes emitting approximately 700 nm, re-

The dc voltage at Point X stabilizes the drive current in the feedback mode against any variations in the IV characteristics of the laser diode and the power-supply voltage.

Figure 1



NOTES:
ALL RESISTORS HAVE 100-ppm STABILITY UNLESS OTHERWISE SPECIFIED.
C_1 IS SILVER-MICA TYPE.

spectively. However, this sensitivity also puts a stringent requirement on the stability of these parameters. The resolution with which the output wavelength can vary depends on the stability or accuracy of the sensitivity parameters. The drive-current sensitivity of 0.025 nm/mA suggests that a 10-MHz accuracy, which is a modest requirement, necessitates a drive-current stability of 0.7 μA, which is equivalent to 7 ppm, assuming a drive current of 100 mA.

The low-cost and small circuit in **Figure 1** is a stable laser-diode driver with an optional modulation-input facility. The circuit features soft start, soft decay, and immunity to noise transients. The circuit operates from a single supply of 12V and uses a quad "Norton" op amp, the LM3900. IC_{1A}, Q_1, R_1 to R_4, and C_1 constitute the basic constant-current source with the magnitude of current depending on the dc voltage present at Point X and the value of R_4. The dc voltage at X stabilizes the drive current in the feedback mode against any variations in the IV characteristics of the laser diode and the power-supply voltage. The feedback signal consists of a proportional voltage appearing across sense resistor R_5, which noninverting IC_{1B} amplifies by a gain 15. The output of this amplifier drives differential amplifier IC_{1C}. One of the inputs to IC_{1C} is a bandgap-derived reference voltage. The differential amplifier has a gain of 15.

A small change in the drive current results in a large change in the control voltage at X in a direction that restores the current to the nominal value. The closed-loop gain of the circuit is approximately 20. Changing the reference voltage to the differential amplifier, which is the voltage at Point Y, changes the nominal value of the current. Although the chosen component values produce a drive current of 60 mA, the circuit can produce drive current of 50 to 80 mA. R_6, C_2, and Q_2 provide soft-start and soft-decay features. The observed soft-start and soft-decay times are approximately 200 and 500 msec, respectively. A lowpass filter comprising R_7 and C_3 has a cutoff of approximately 10 Hz in the feedback loop to provide immunity to fast transients.

Tests show that the circuit has a stability better than ±0.05%/hour. The observed short-term current stability is better than ±0.02%. The observed variation in drive current for a ±2V variation in power-supply voltage is less than 0.1%. Experimental measurements by connecting an appropriate resistance across R_4 introduce a step change of 2 mA. Measurements also show the resultant change in current and a closed-loop gain of approximately 20. The Toshiba (www.toshiba.com) TOLD-9211 laser diode emitting approximately 4 mW tested the circuit for a drive current of 60 mA at 670 nm. The circuit fits into a DIP-like, eight-pin metal package. (DI #2291)

To Vote For This Design,
Circle No. 501

Voice-storage chips talk to each other

Jerzy Chrzaszcz, Warsaw University of Technology, Poland

Of the many solid-state voice-storage chips available, the ChipCorder family from Integrated Storage Devices (ISD, San Jose, CA) is one of the most user-friendly. A single chip integrates nonvolatile voice memory, a microphone preamplifier, and an output stage capable of driving a 16V loudspeaker. A simple interface allows you to record and play messages under manual or μP control. The configuration in **Figure 1** allows you to copy the contents of one chip to another. For single units, consider recording each chip anew. For regular production, you could purchase a gang programmer from ISD. However, for prototyping and short production runs, the circuit in **Figure 1** offers an attractive cost/performance ratio.

The programmer accommodates 25-xxx-series chips with recording time as fast as 120 sec. It consists of two ZIF sockets, control logic, and some passive components. The output signal from the Master chip traverses R_1, R_2, C_1, and C_2 to the Target inputs. R_3 and C_3 couple the Target's preamplifier and amplifier. R_4 and C_4 provide an AGC delay to the Target (Consult the ISD data sheet for details.) Strapping of the control pins ensures that the Master can only play back and the Target can only record; however, to avoid hazardous transient states, you should lock the voice chips in their sockets before switching on the power.

The controller is configured as an asynchronous state machine that uses just two 7474 flip-flops. Its simplicity results from the highly autonomous operation of the voice chips. This design uses the M4 function mode (A4, A8, and A9 pulled high), which provides sequential addressing of the messages without controller intervention. Closing the Copy switch starts Master playback simultaneously with Target record (CE set low, LED1 on). When the Master issues "End of Message" (EOM), recording stops (CE set high, LED1 off) and the EOM marker automatically goes into the Target's memory. The cycle repeats whenever you close the Copy switch, so you can copy messages one by one.

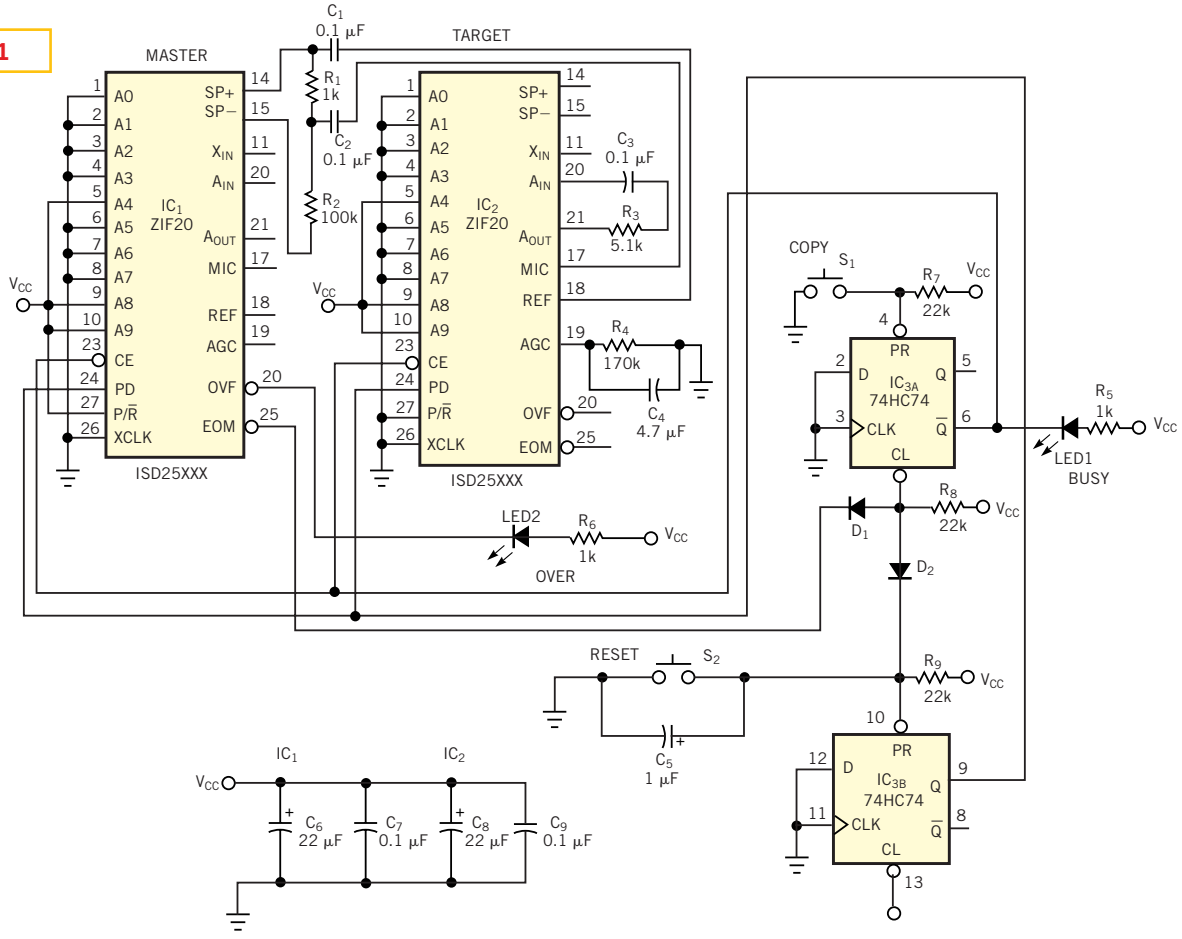
After you copy the entire Master contents, the overflow (OVF) line goes low, signaling an overflow condition. This signal turns LED2 on. Closing the Reset switch clears the flip-flop and resets the internal address counters of the voice chips. The capacitor across the Reset switch generates a power-on reset pulse. The circuit uses internal clocking for the voice chips; therefore, actual message

addressing may differ slightly from copy to copy. This effect is negligible, as long as all messages fit into memory and your application uses a sequential-access mode. However, even if your target system directly addresses the chip, you can still use this programmer if the Master copy provides some spacing between consecutive messages. You thus avoid the potential errors of overlapping EOM markers past the nominal start address of the next message. Also, because of AGC regulation, the sound level of Target messages may, and probably will, differ from the original. (DI #2272).

To Vote For This Design,
Circle No. 502

You can avoid the expense of a gang programmer by using this simple scheme to copy voice-chip messages.

Figure 1



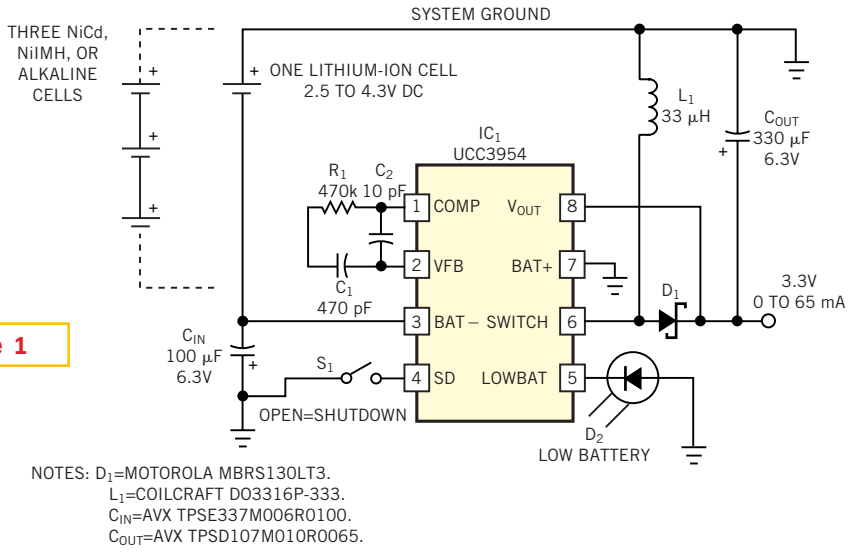
3.3V converter delivers 3W from Li-ion battery

David Salerno, Unitrode Corp, Merrimack, NH

Lithium-ion batteries are rapidly gaining popularity in portable applications because of their superior energy density, low self-discharge rate, and high cell voltage. When you use one Li-ion battery to power a 3.3V dc/dc converter; however, you encounter a problem, because the battery voltage can be higher or lower than 3.3V. When fully charged, a Li-ion cell has approximately 4.2V output; when fully discharged, the voltage is approximately 2.5V. Therefore, you cannot use a simple buck or boost topology with a single inductor to generate a regulated 3.3V output. Some designs boost the voltage to approximately 4.3V and then use a low-dropout regulator to produce the 3.3V. This approach is inefficient, and efficiency is a crucial consideration in battery-powered applications. The circuit in **Figure 1** offers a solution to the problem.

The circuit works by referencing the

positive terminal of the battery to system ground and using a flyback topology with a single low-cost inductor to generate 3.3V, with respect to system ground. IC_1, a UCC3954, is a fixed-frequency, 200- kHz voltage-mode PWM converter that includes an internal 0.15V MOSFET switch. Gate drive for the FET comes from bootstrapping off the 3.3V output. The converter works efficiently over a load of 0 to 650 mA. Note that the input and output filter capacitors should be low-ESR tantalums or OSCONs. Output ripple is lower than 1% at maximum load. The inductor value is not critical; 33 mH is a good compromise between size and efficiency.



NOTES: D_1=MOTOROLA MBRS130LT3.
L_1=COILCRAFT DO3316P-333.
C_{IN}=AVX TPSE337M006R0100.
C_{OUT}=AVX TPSD107M010R0065.

Figure 1

This 3.3V dc/dc converter takes full advantage of the benefits of an Li-ion battery and works over the battery's full range of 4.2 to 2.5V.

The compensation components (R_1, C_1, and C_2) ensure stability and provide good transient response over a wide load. For applications in which no sudden changes in load current occur, you can use a simpler, dominant-pole compensation method. In this case, you can omit R_1 and C_1 and increase C_2 to 0.039 µF. The UCC3954 includes a low-battery-warning output and a shutdown input. The low-battery warning is a current-limited, open-drain output that turns on when the battery voltage approaches the shutdown threshold of the IC. You can use it to turn on an LED or to drive an input to a µP to provide an alert that power will soon be lost.

To enable IC_1, you should pull the shutdown input up to output ground. When this input is left open, it pulls down to the battery (–) potential, and IC_1's quiescent current reduces to less than 1 mA. To prevent overdischarging the Li-ion battery, IC_1 automatically turns off when the input voltage drops to less than 2.5V, and the quiescent current reduces to 30 mA. Although IC_1 is designed for use with single-cell Li-ion batteries, you could also power the converter using three nickel-based rechargeables or three alkaline cells in series. As with any high-frequency converter, layout and grounding critical to proper operation. Keep all connections as short as possible, and use a ground plane. (DI #2263).

To Vote For This Design,
Circle No. 503

Pulsing charge pump drives capacitive loads

Paul J Rose, Mental Automation Inc, Bellevue, WA

The test circuit in **Figure 1** efficiently drives various capacitive loads, such as memory cells and simple capacitors, so that you can observe their leakage effects. Essentially, the circuit is a pulsed and variable current source acting as a charge pump. A pulsed voltage source drives a one-shot oscillator. This one-shot drives two MOSFET switches that convert the 10V rail-to-rail output of the oscillator to the desired rail-to-rail voltage drive—in this case, 15V—for the controlled current mirror with the same voltage- switching polarity. The current mirror drives the variable load.

R_1 and C_1 determine the timing pulses that IC1's one-shot oscillator produces. When IC1's output is high, Q_1 is on, and Q_2 is off. The floating drain of Q_2 causes the emitter and base of Q_3 to have the same potential, so that Q_3 is off. Then, the pnp current mirror of Q_4 and Q_5 turns on to drive the variable load of R_2 and C_2 high. R_3 controls the charge rate of the load. As you make R_3 smaller, the current mirror provides more current to the load to charge it up faster, as required for testing.

When the output of the one-shot is low, Q_1 is off, and Q_2 is on. In this case, the drain of Q_2 is at ground potential, and the base of Q_3 is at a lower potential than its emitter so that Q_3 turns on. Current through Q_3 flows through R_3, causing a voltage rise at the bases of Q_4 and Q_5, which turns them off. Turning off Q_4 and Q_5 disconnects the variable load from its power supply so that the load is free to bleed stored charge through R_2.

This pulsing charge pump has three unique features: It can generate various pulse widths, the variable resistor in the coupled-collector circuit of the pnp cur-

rent mirror provides a variable charging rate, and the circuit accommodates separate voltage drives for the one-shot oscillator and load using MOSFET switches. Any signal-propagation delay or asymmetrical switching effects through the pump cause no adverse latency effects if the pulse periods are on the order of 100 msec or more. Note that you can replace Q_1 and Q_2 with a variable gain buffer follower as **Figure 1** indicates. In this case, one 15V supply drives the follower.

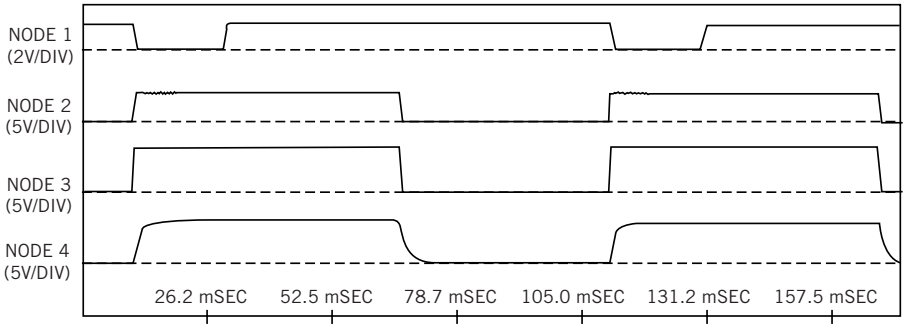
Spice simulations, using models developed in-house and by semiconductor vendors, confirm the circuit's operation (**Figure 2**). (DI #2248)

To Vote For This Design, Circle No. 504

Acknowledgment

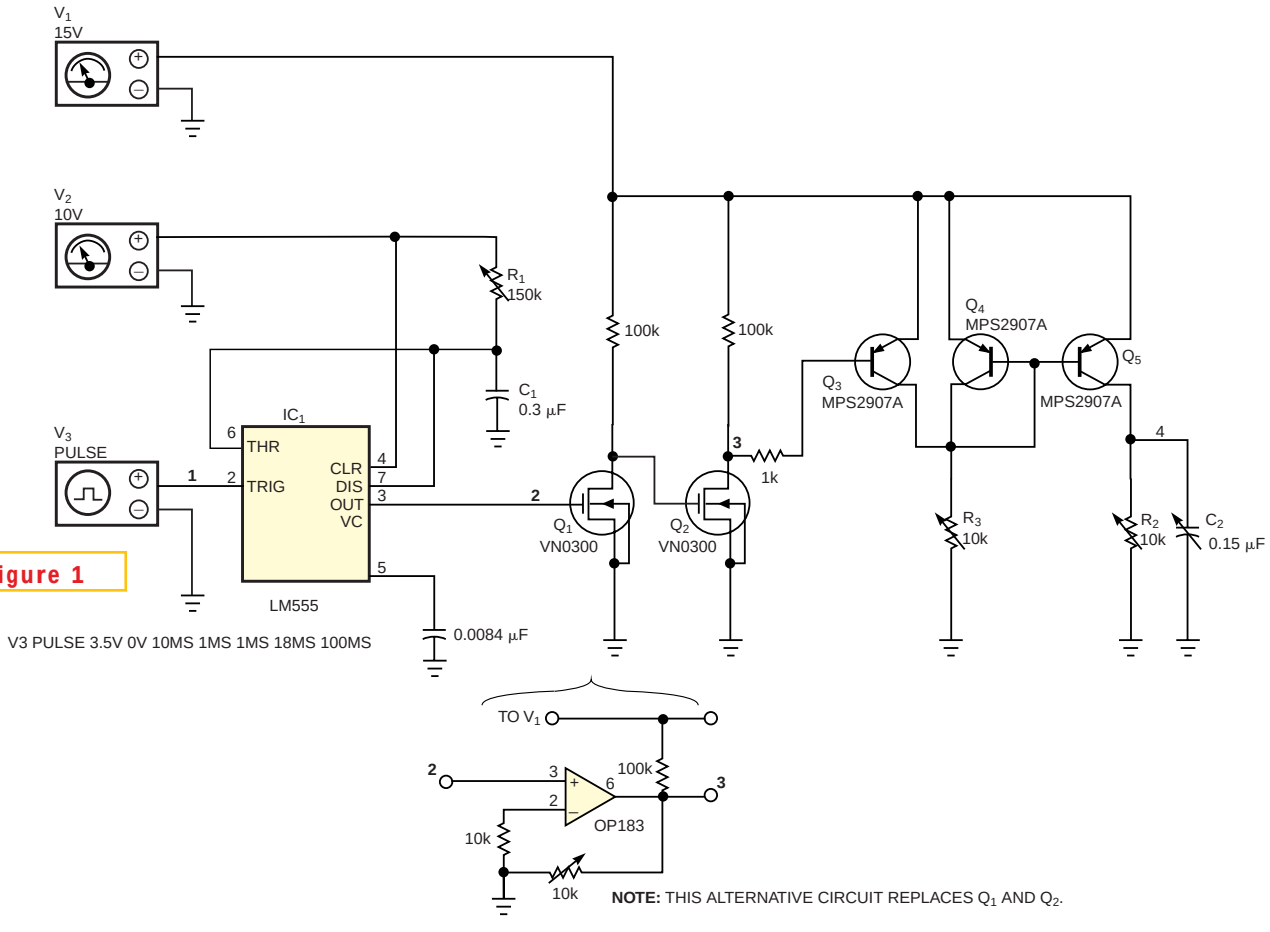
The author thanks Mental Automation Inc, whose ECAD tools he used to implement and test the circuit. The company provided the author the company time to submit this idea. The author also thanks Seattle Silicon Inc (Bellevue, WA) for permitting him to publish this Design Idea, which the author built and tested in Seattle Silicon's laboratory as part of a test project.

Figure 2



Spice simulations illustrate the waveforms at nodes 1 through 4 in the pulsing charge-pump circuit.

Figure 1



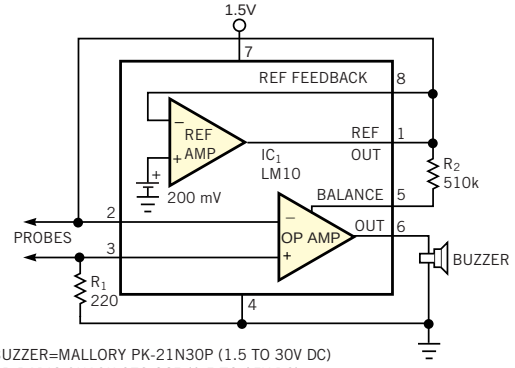
This pulsed and variable current source acts as a charge pump to efficiently drive various capacitive loads.

Short-circuit finder uses few parts

Boris Khaykin, Candid Logic Inc, Madison Heights, WI

The simple tester in **Figure 1** detects short circuits on assembled pc boards and also rings out cables and harnesses. The short finder has a narrow zone of threshold uncertainty and very low "insertion" voltage and current, and it's not confused by capacitors. The circuit uses an LM10, an IC that combines a precision 200-mV reference, a reference buffer, and an independent, high-quality op amp. It can operate from supply voltages of 1.1 to 40V. The op amp in this design serves as a comparator. The voltage from the reference buffer, via R_2, creates a positive-going bias shift at the balance input and a negative-going bias shift at the comparator's inverting input.

Figure 1



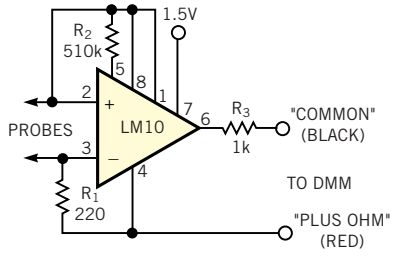
NOTES: BUZZER=MALLORY PK-21N30P (1.5 TO 30V DC) OR RADIO SHACK 273-065 (1.5 TO 15V DC).

Keep an ear open for short circuits, with this easy-to-build short-circuit tester.

When the tested circuit resistance exceeds 2V, the negative-going bias overrides the positive-going bias, and the comparator delivers 0V to the buzzer. Otherwise, the comparator delivers full output voltage to the buzzer to indicate a short circuit. R_1 limits the current to the circuit under test to less than 1 mA. The circuit's current drain is less than 300 µA with open test probes and approximately 2 mA with the probes shorted together. Open-circuit voltage is 200 mV, which is less than the turn-on voltage for pn junctions. If desired, you can set the voltage as low as 15 mV by adding 18Ω resistance between pins 2 and 3 of IC_1. However, the quiescent current increases to 1 mA.

You can change the resistance threshold by changing the value of R_2. With the values shown, the threshold is approximately 2V. The supply voltage can be within 1.1 to 30V, depending on the buzzer's voltage range. You can use any piezo buzzer with current consumption lower than 20 mA. You can easily build the short finder as an adapter for a DMM, provided that the DMM has a continuity function (**Figure 2**). Upon detection of a resistance that is less than 2V, the short finder delivers a virtual negative resistance to the DMM. By nature, this signal is lower than any DMM continuity threshold (which is always positive); therefore, the circuit works with any DMM. R_3 limits the current to the DMM's input circuitry to approximately 1 mA. (DI #2264).

Figure 2



Add a short-circuit test capability to your DMM, using this modification of the circuit in Figure 1.

To Vote For This Design, Circle No. 505

"Tube" circuit provides linear tuning

Lyle Williams, Electronic Technical Services, New Orleans, LA

Parallel LC circuits that you tune by changing capacitance have a nonlinear frequency-versus-voltage or frequency-versus-shaft-position characteristic. The frequency of an analog-tuned circuit is proportional to the reciprocal of the square root of the tuning capacitance. When you tune a bandwidth that is say, 5% or less of the center frequency, the frequency-versus-capacitance over this limited band is essentially linear. Because the frequency is proportional to capacitance, it's desirable to have a linear capacitance-versus-shaft-position or capacitance-versus-voltage characteristic. A mechanical variable capacitor can provide a linear capacitance-versus-rotation characteristic. However, mechanical tuning capacitors are expensive and large and have limited reliability.

You frequently use varactor diodes for voltage control of capacitance. But their capacitance-versus-voltage characteristic is approximately logarithmic, not linear. In the days of vacuum tubes, designers used reactance-tube circuits for automatic frequency control in FM receivers and for

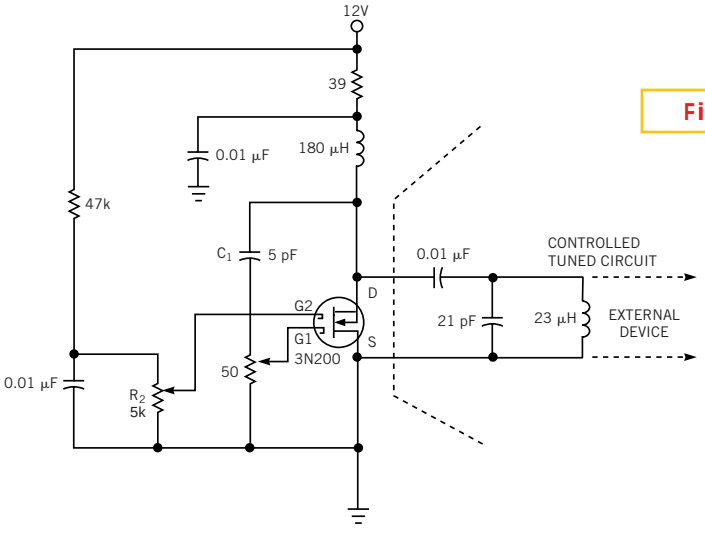


Figure 1

Use Grandpa's circuit without a filament to heat the room for linear frequency tuning in a regenerative radio receiver.

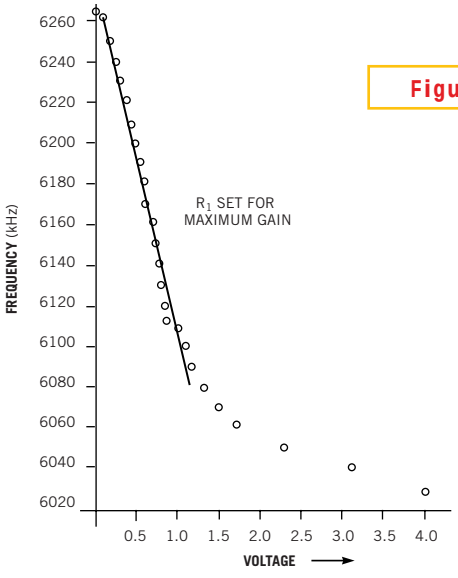


Figure 2

The frequency-versus-voltage characteristic is linear when you use voltage control in Figure 1's circuit.

modulating FM transmitters. It's possible to make the capacitance of the circuit proportional to the transconductance (g_m) of the tube. Over a certain bias range, the tube's g_m is proportional to the grid bias voltage. You can build such a reactance circuit using FET or bipolar transistors (**Figure 1**). The current in the drain circuit is in quadrature with the drain voltage because of the feedback elements R_1 and C_1. As a result, the drain circuit emulates a capacitor.

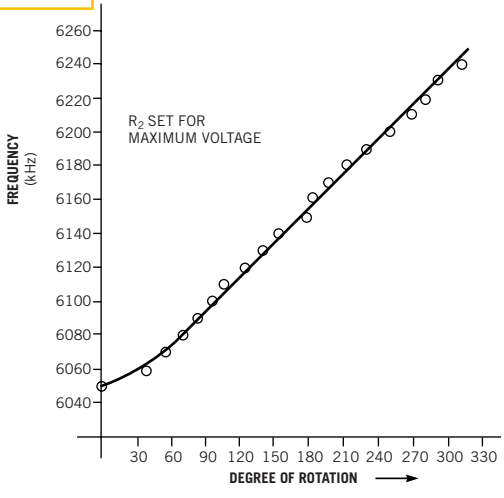
You can control the capacitance using voltage, via potentiometer R_2, or by adjusting R_1's shaft position. (You should set the unused potentiometer to maximum.) The L and C values in this controlled tuned circuit are chosen for a frequency range in the vicinity of the 49m short-wave band. The controlled LC circuit serves to tune a regenerative-type receiver. The tuning dial for this radio is linear—a feature uncommon in analog receivers. A modern version of the regenerative receiver can provide performance comparable with that of a simple superheterodyne receiver. Regenerative receivers are unique in that they require only one LC resonant tuning circuit. A superhet requires at least two resonant circuits that must track each other as you tune the receiver.

You can change the frequency band of a regenerative receiver by switching a single two-terminal inductor. You could also use **Figure 1**'s controlled tuned circuit to tune an RF amplifier, a filter, or an oscillator. The reactance circuit produces a maximum capacitance of $C_R = g_m \times R_1 \times C_1$. R_1 is the total resistance of potentiometer R_1. The reactance of C_1 should be much larger than R_1 at the frequency of interest: $X_{C1} >> R_1$. **Figure 2** shows the result of using voltage tuning via potentiometer R_2. The curve is linear from 0.1 to 1.3V. The change in frequency that accrues in this voltage range is 6.2 to 6.07 MHz for a 190-kHz bandwidth.

Figure 3 shows the results of shaft tuning. The bend at the lower end of the curve comes from the potentiometer characteristic. The curve is linear throughout the entire tuning range, which is 210 kHz wide. In the reactance-"tube" circuit, it's desirable to use a transistor with high output impedance. In this respect, a pentode vacuum tube with an output impedance of approximately 750 kV is superior to a transistor. However, MOSFETs have a considerably higher g_m than tubes. The transistor's g_m determines the amount of change in capacitance that is possible. The maximum g_m of a 3N200 MOSFET is 15,000 µmho, and the output impedance is 13 kV. (DI #2267).

To Vote For This Design,
Circle No. 506

Figure 3



A rare feature in analog receivers, shaft (dial) angular position is linear with respect to frequency.

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____

E-mail _____ **Fax** _____

Company _____

_____

Address _____

_____

Country _____ **ZIP**_____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

design ideas

Edited by Bill Travis and Anne Watson Swager

Voltage reference sets current limit

Joe Engle, Burr-Brown Corp, Tucson, AZ

Power op amps have a real need for active output-current limiting. Most power-amplifier designs rely on the voltage drop across a user-supplied sense resistor to turn on an internal transistor. This method has several drawbacks, notably, an inability to change the current-limit point under program control. The current-limit circuit in **Figure 1** allows you to establish the setpoint by applying a voltage to one of the amplifier's pins. With this design, it is possible to set the current-limit point with the output of a DAC, possibly under the control of an embedded µC.

The OPA547 is a true op amp; thus, it does not need a connection to power ground. The current-limit-setting voltage for this IC uses the negative supply as a reference. For single-supply applications in which the negative supply is ground, this referencing technique presents no problem, but for circuits that use a negative supply below ground potential, you need a different technique. The circuit in **Figure 1** shifts the reference potential for the control signal from ground to the negative supply. For simplicity, **Figure 1** shows the OPA547 as an inverting amplifier, but you can use any op-amp application circuit. The circuit uses an OPA340 for reference shifting because it is capable of rail-to-rail operation on both input and output.

To understand the operation of the reference-shifting circuit, first recognize that the R_3-to-R_4 voltage divider sets the voltage at IC_2's Pin 3. Thus, the intermediate voltage (V_i), as measured from the negative supply, is given by

$$V_i = V_C \frac{R_4}{R_3 + R_4}. \qquad (1)$$

To find the voltage at IC_2's pin 2, note that the current through R_3 equals the sum of the currents in R_1 and R_5, leading to the following expression:

$$\frac{V_1 - V_i}{R_2} = \frac{V_i - V_O}{R_1} + \frac{V_i}{R_5}. \qquad (2)$$

As long as op amp IC_2 operates in the linear region, the voltage at Pin 2 equals the voltage at Pin 3, so the value of V_i in each of the expressions is equal. When you substitute the first term into the sec-

I (mA)	V_C
0	4.75
100	4.12
375	2.38
500	1.59
750	0.01

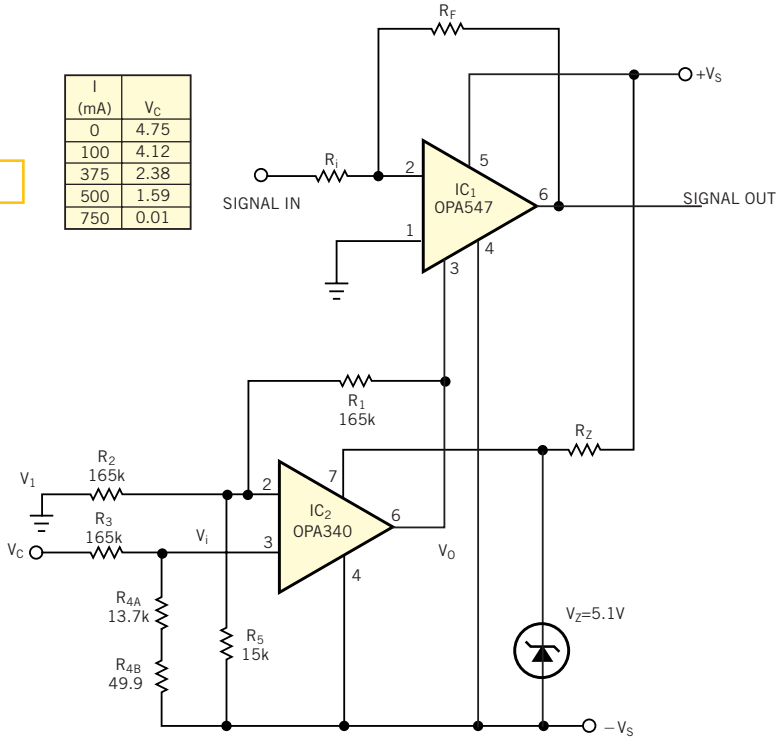


Figure 1 You can use a difference amplifier with wide common-mode range to control a power amplifier's current limit.

ond, set R_1 equal to R_2, and combine terms, the resulting expression is

$$V_O = V_C\left(2 + \frac{R_1}{R_5}\left(\frac{R_4}{R_3 + R_4}\right)\right) - V_1. \quad (3)$$

In **Figure 1**, V_1 connects to ground, and you obtain unity gain by setting the coefficient of V_C in **Equation 3** to 1. If you expand and combine terms, the expression becomes

$$1 + \frac{R_1}{R_5} = \frac{R_3}{R_4}. \quad (4)$$

To change the scalar relationship between the controlling voltage applied to the power op amp, simply set the coefficient term to the desired value and solve **Equation 3**. To determine the resistor values, consider the worst-case common-mode voltage that IC_2 can encounter. OPA547 allows a maximum supply differential of 60V. In an extreme case, the positive supply of the OPA547 connects to ground and the current-limit set voltage is +5V. **Equation 1** becomes

$$5 = 65\frac{R_4}{R_3 + R_4}, \quad (5)$$

which reduces to $R_3=12R_4$. Applying this ratio to **Equation 4** and setting R_3 equal to R_1 produces $R_1=11R_5$. Selecting from a list of standard 1% resistor values yields the values in **Figure 1**. Note that the stage operates with a common-mode voltage that equals the negative supply. Errors in the resistor values can produce a significant offset shift. With this circuit, it is possible to set the current limit of the power op amp to a known, repeatable value under program control. (DI #2270).

To Vote For This Design,
Circle No. 406

Accelerometer output gives temperature info

Harvey Weinberg, Analog Devices Inc, Cambridge, MA

The adxl202 dual-axis micromachined accelerometer from Analog Devices (Norwood, MA) is appropriate for high-resolution applications. In these applications, you sometimes need to know the ambient temperature for control purposes or for circuit-drift compensation. The scheme in **Figure 1** offers a novel way to convey temperature information to the system µC without the need for an A/D converter or any additional I/O pins. The ADXL202 delivers two PWM signals that are proportional to the acceleration in its X and Y axes. Current in the R_{SET} resistor sets the period of the PWM signals.

You can use a thermistor in series with or instead of the R_{SET} resistor to vary the PWM period with temperature. However, because of the grossly nonlinear response of thermistors, the PWM period is also grossly nonlinear with temperature. In addition, the thermistor's poor sensitivity at high temperatures may be unacceptable. Although R_{SET} normally connects to ground, you can connect it to any noise-free voltage source ranging from 0 to approximately 1.2V (at which voltage the internal current source runs out of compliance). By connecting R_{SET} to the V_{OUT} pin of a TMP36 temperature sensor, the PWM-period set current varies fairly linearly (within ±5°C) with temperature from –20 to +40°C. Therefore, the PWM period varies linearly with temperature. You can easily extract temperature information from the PWM signal, because you normally measure the period to determine duty cycle. (DI #2271).

To Vote For This Design,
Circle No. 407

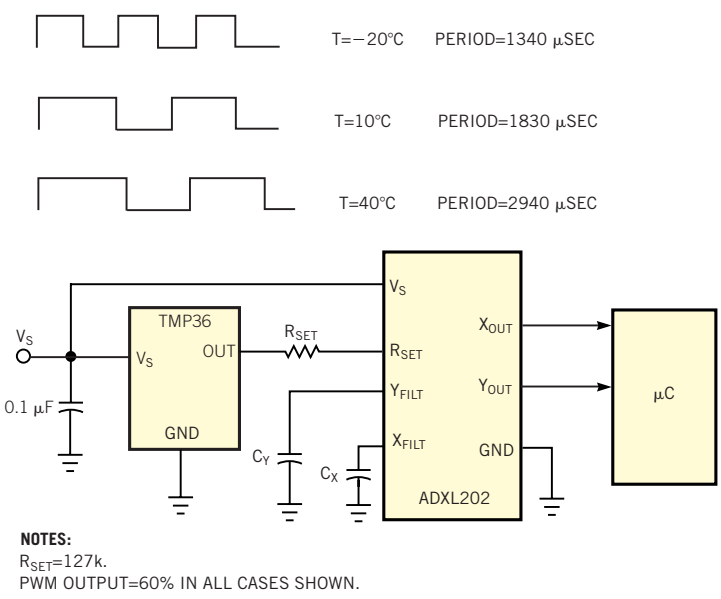


Figure 1

An accelerometer can do double duty, by supplying both acceleration and temperature information.

Voltage comparator forms pulse demodulator

Abel Raynus, Armatron International, Melrose, MA

To process low-level ultrasound or radio-range pulses, you need a signal-conditioning amplifier followed by a pulse demodulator to translate the signals to dc pulses. Traditionally, you would use a diode-demodulator configuration (for example, the circuit in **Figure 1a**) with one stage of a single-supply op amp. The circuit in **Figure 1b** does the same job but uses a voltage comparator instead of a diode demodulator. The key to the method is choosing a threshold voltage (V_{TH}) on the negative comparator input that is slightly higher than the dc level of the amplifier output, which is equal to or close to $V_{CC}/2$. The R_4-to-R_6 resistive divider determines the difference between the op-amp bias and the threshold voltage. This difference, calculated to yield an acceptable signal-to-noise voltage, is $V_{CC}R_5/(R_4+R_5+R_6)$.

Or, assuming $R_4=R_5+R_6$, the difference is $V_{CC}R_5/2R_4$. R_3 and C_2 make up a lowpass filter. This pulse demodulator has some advantages: First, its sensitivity is higher than that of a diode demodulator. A 25-mV, 40-kHz, 1-msec input pulse produces a 0.1V output pulse in **Figure 1a**'s circuit, and a 2V output pulse in **Figure 1b**'s circuit. Second, it's convenient and economical to use one more stage of the dual or quad op amp instead of adding discrete components. (DI #2273).

To Vote For This Design,
Circle No. 408

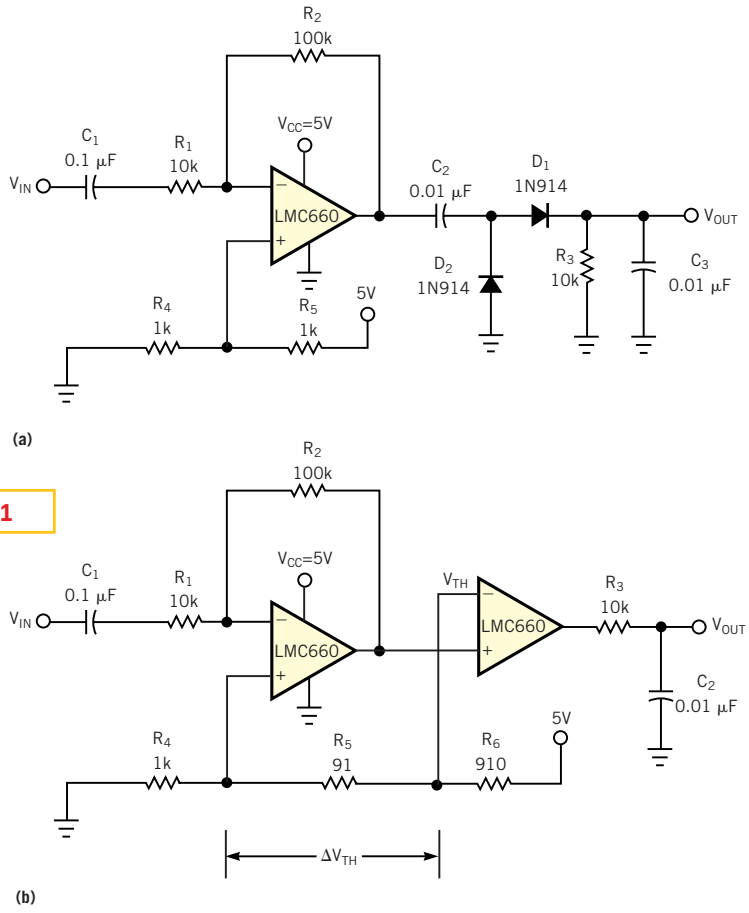


Figure 1

The passive, diode-based approach to pulse demodulation (a) provides a lower sensitivity than the active approach (b).

Watchdog-reset catcher aids embedded-system debugging

Scott Newell, PCSI, Fort Smith, AR

A simple "junk-box" circuit uses a 4013 CMOS flip-flop and a handful of passive components to determine whether random resets are the result of a blown stack or the result of the watchdog-reset circuit tripping (**Figure 1**). You can also use this circuit to "grab" and hold other logic level edges like memory or I/O accesses.

A logic-level rising edge at the clock input (Pin 3) of the 4013 clocks the flip-flop. Because the circuit holds the data input (Pin 5) high, the Q output (Pin 1) goes

high, which turns on the LED. Once the LED is on, the circuit ignores any further changes at the input.

R_1 and C_1 are the power-up reset for the flip-flop. At power-up, C_1 discharges, which holds the reset input (Pin 4) of the 4013 high, clears the Q output of the 4013, and turns off the LED. C_1 charges up to the supply voltage through R_1, taking the R input (Pin 4) low to deassert the 4013 reset time. D_1 discharges C_1 quickly on power-down. S_1 is an optional reset switch. C_2 is a power-supply bypass capacitor. Don't forget to ground all unused inputs on the 4013. To reset the circuit, either momentarily close S_1 or temporarily disconnect power.

You can solder all the parts onto a BNC, which makes it easy to connect a scope probe directly to the watchdog-reset catcher. You can use a clip lead for the power line and easily steal power from the device under test. You can then connect the output of the embedded system's watchdog-reset circuit through the scope probe to the clock input of the 4013.

None of the part values are critical, and many types of flip-flops can substitute for the 4013. A faster flip-flop may be necessary to watch fast signals. Adding an inverter to the input would allow you to catch falling edges, such as active-low reset signals. (DI #2293)

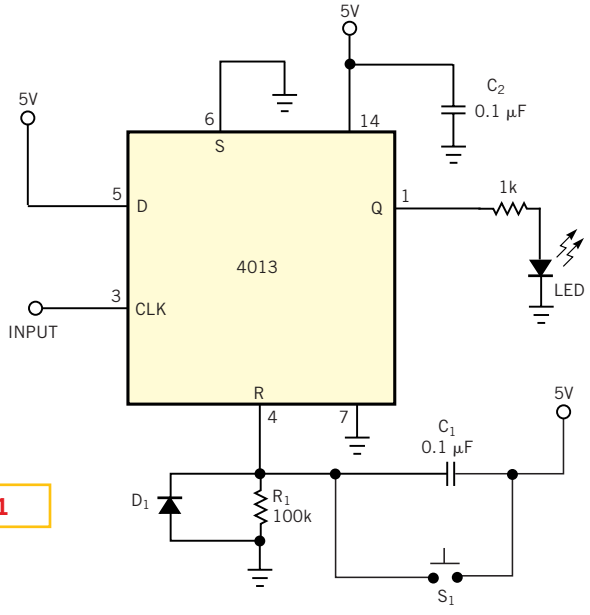


Figure 1

NOTES:
LED=HIGH-EFFICIENCY RED LED.
D_1=IN4001 OR ANY OTHER SMALL-SIGNAL DIODE.
S_1=OPTIONAL MOMENTARY PUSHBUTTON SWITCH.
TIE UNUSED INPUTS TO GROUND.

A 4013 CMOS flip-flop and a handful of passive components monitor the activity of an embedded system's watchdog reset.

To Vote For This Design, Circle No. 409

Easily display bit-map images on small-graphic LCDs

Todd Fitzsimmons, Densitron Corp, Santa Fe Springs, CA

Combining a high-level language, such as Microsoft Windows, with low-level assembly code or C++ code allows you to display perfect bit-map pictures on your small-graphic LCD screen. You can use Microsoft Paint or any other bit-map-generating program to define and edit the picture. Unlike segmented and alphanumeric LCDs, small-graphic LCDs are fully graphical and can display logos, graphs, or any other image in addition to numbers and characters. The main hurdle to upgrading to this type of display is the software necessary to display the bit-map pictures.

To create your bit maps, you can use any available bit-map program, such as Microsoft Paint. After opening this program, select the Attribute menu. At the prompt, you enter the LCD size you are using and then choose the monochrome-bit-map option. You can now close the Attribute window and start drawing, typing, or pasting the images you want. After saving the pictures, you need to attach the bit maps to the end of your assembled or compiled program in order of their intended use. You can attach them using the COPY/B command in DOS, which differs from the regular COPY command by copying directly as a binary format without adding a byte of data at the end.

To use these attached bit maps, you need a subroutine in your program, such as the 8051 assembly code in **Listing 1**, which can pull the data out of the lower memory and send it to the display. (You can download this listing from *EDN*'s Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2295.) This code clocks in bit maps to the SED1330 controller chip on a 240×320-pixel display. If your display doesn't use the SED1330 chip, you can still use this code with modifications to the WRCMD and WRDATA subroutines and possibly some alterations on the direction you clock in the data. The di-

rection in the SED1330 is the same as the T6963 controller chip. However, if you use the HD61830, you need to switch the direction of the data; D_7 becomes D_0, D_6 becomes D_1, and so on.

For a bit-map program, the first 62 bytes of data call out the protocol for the rest of the bit-map code, such as the type, size and layout of the bit map. Because you have selected the monochrome option and specified layout dimensions using a bit-map-generating program, the subroutine can skip the first 62 bytes. The 63rd byte defines the first 8 pixels in the lower left corner of the display. The following bytes go sequentially to the screen until you hit the right edge of your display. The next byte is either the first byte on the next row up on the left side or a padded zero that the bit-map program places there to maintain certain integers for row length.

Padded zeros are necessary when the number of bytes in a row are not divisible by four. If you have 16 bytes of data per row, no padded zeros are necessary. However, if there are 30 bytes per row, two padded zeros are necessary to bring the number of bytes to 32. Your internal program must disregard these zeros before going on with the 33rd byte of data (**Table 1**).

Consider the example of driving a 128×240-pixel display. You would set up your assembly code to strip off and discard the first 62 bytes of data from the bit-map file. The 63rd byte is then the first byte in the lower left of the LCD. Then, the next 29 bytes of data (240/8=30) appear directly in the display. The code must then discard the next 2 bytes of padded zeros. The next byte of data then appear in the next row up and over on the left. A user continues this process until all 128 lines are completed.

If you access the upper bit-map memory by using the data pointer address in your µP, then, when you paint the first page and increment the data pointer, you see the first byte of the next picture in your list.

An important difference between a bit map and an LCD is that, in bit-map programs, a binary 1 is an off pixel, and a binary 0 is an on pixel. So, a user must perform an exclusive-OR with FFh to properly view the bytes. Without this operation, your picture would be the inverse image of your original picture. (DI #2295)

TABLE 1—CORRELATION BETWEEN BIT-MAP RESOLUTION AND PADDED ZEROS

Bit-map resolution (pixels)	Bytes per row	Padded zeros per row	Totals divisible by four
32×80	10	2	12
32×202	26	4	30
33×100	13	3	16
64×128	16	0	16
64×240	30	2	32
64×480	60	0	60
128×128	16	0	16
128×240	30	2	32
128x×56	32	0	32
200×640	80	0	80
240×320	40	0	40

To Vote For This Design,
Circle No. 410

LISTING 1—BIT-MAP-EXTRACTION SUBROUTINE

```
;.....
;
;  This is a base program for clocking in bitmaps to the SED1330 on a 240x320 LCD display
;  using 8051 assembly code.
;
;.....
                ;
START:          MOV   30H,#3EH          ;SETS UP THE INNER LOOP TO 62
                MOV   31H,#01H          ;SETS UP FOR NULL DATA
                LCALL CK_DATA           ;GO GET DATA
                MOV   B,#0F0H           ;SETS UP INNER LOOP FOR 240 LINES OF DATA
                MOV   32H,#008H         ;SETS LOWER CURSOR ADDRESS (address where program ends/bitmap begins)
                MOV   33H,#02AH         ;SETS UPPER CURSOR ADDRESS
DO_OVER:        PUSH  B                 ;STORE VALUE
                MOV   30H,#28H          ;SETS UP THE INNER LOOP TO 40 CHARACTERS PER LINE
                MOV   31H,#00H          ;SETS UP FOR VALID DATA
                LCALL SET_CUR           ;SETS THE INITIAL ADDRESS ON DISPLAY
                LCALL CK_DATA           ;GO GET DATA
                POP   B                 ;RECALLS INNER LOOP
                DJNZ  B,CONTINUE        ;IF NOT ZERO, THEN REDO (moving up one line on display if doing over)
                RET                     ;
CONTINUE:       MOV   A,0D0H                    ;RECALL CARRY FLAG
                ANL   A,#7FH            ;CLEAR THE CARRY FLAG
                MOV   0D0H,A                    ;STORE FLAG
                MOV   A,32H             ;RECALL LOWER CURSOR ADDRESS
                SUBB  A,#28H            ;SUBTRACT 40 FROM THIS ADDRESS TO MOVE UP ONE LINE
                MOV   32H,A             ;RESTORE THE NEW LOWER ADDRESS
                MOV   A,0D0H            ;RECALL CARRY FLAG
                ANL   A,#80H            ;STRIP OFF CARRY FLAG
                CJNE  A,#80H,NO_CARRY   ;SEE IF FLAG IS SET (if set then must bump down higher address by 1)
                MOV   A,33H             ;RECALL UPPER CURSOR ADDRESS
                SUBB  A,#01H            ;SUBTRACT 1 FORM UPPER
                MOV   33H,A             ;RESTORE THE VALUE
NO_CARRY:       LJMP  DO_OVER           ;DO INNER ROUTINE OVER AGAIN
CK_DATA:        MOV   A,31H             ;RECALL BYTE FOR NULL OR VALID DATA
                CJNE  A,#00H,GET_DATA   ;SEE IF NULL DATA OR NOT (selects appropriate subroutine)
DATA_IN:        LCALL DATA_ARM          ;GET THE VALID DATA
                LCALL WRCHAR            ;SEND IT TO THE DISPLAY
                DJNZ  30H,DATA_IN       ;COMPARE LOOP AND DECREMENT UNTIL ZERO
                RET                     ;
GET_DATA:       LCALL DATA_ARM          ;GET THE NULL DATA
                DJNZ  30H,GET_DATA      ;SEE IF INNER LOOP IS ZERO
                RET                     ;
SET_CUR:        MOV   B,#46H                    ;SET ADDRESS POINTER
                LCALL WRCMD             ;CLOCK IN
                MOV   B,32H             ;RECALL THE LOWER CURSOR POSITION
                LCALL WRDATA            ;CLOCK IN
                MOV   B,33H             ;RECALL THE HIGHER CURSOR POSITION
                LCALL WRDATA            ;CLOCK IN
                RET                     ;
DATA_ARM:       MOV   82H,34H           ;MOV LOWER DATA POINTER ON THE STACK
                MOV   83H,35H           ;MOV HIGHER DATA POINTER ON THE STACK
                MOV   A,#00H                    ;CLEAR OUT REGISTER FOR TRUE DATA POINTER
                MOVC  A,@A+DPTR         ;GET THE DATA POINTED TO BY THE DATA POINTER
                MOV   B,A               ;SWAP DATA TO "B" REGISTER
                INC   DPTR              ;INCREMENT THE DATA POINTER
                MOV   34H,82H           ;RESTORE LOWER DATA (includes increment)
                MOV   35H,83H           ;RESTORE HIGHER DATA
                RET                     ;
WRCHAR:         PUSH  B                 ;STORE DATA
                MOV   B,#042H           ;SETS WRITE DATA COMMAND
                LCALL WRCMD             ;CLOCK IN
                POP   B                 ;RECALL DATA
                XRL   B,#0FFH           ;INVERSE DATA TO PREVENT INVERSE VIDEO
                LCALL WRDATA            ;GO CLOCK IN
                RET                     ;
WRDATA:         MOV   0C0H,#03H         ;SETS WR AND RD HIGH AND A0 LOW
                MOV   0C0H,#01H         ;STROBE WR LOW
                MOV   090H,B                    ;PUTS DATA ON PORT 1
                MOV   0C0H,#03H         ;STROBE WR HIGH
                RET                     ;
WRCMD:          MOV   0C0H,#07H         ;SETS WR, RD, AND A0 HIGH
                MOV   0C0H,#05H         ;STROBE WR LOW
                MOV   090H,B                    ;PUTS COMMAND ON PORT 1
                MOV   0C0H,#07H         ;STROBE WR HIGH
                RET                     ;
                END
```

CMOS inverter VCO tunes octave to UHF

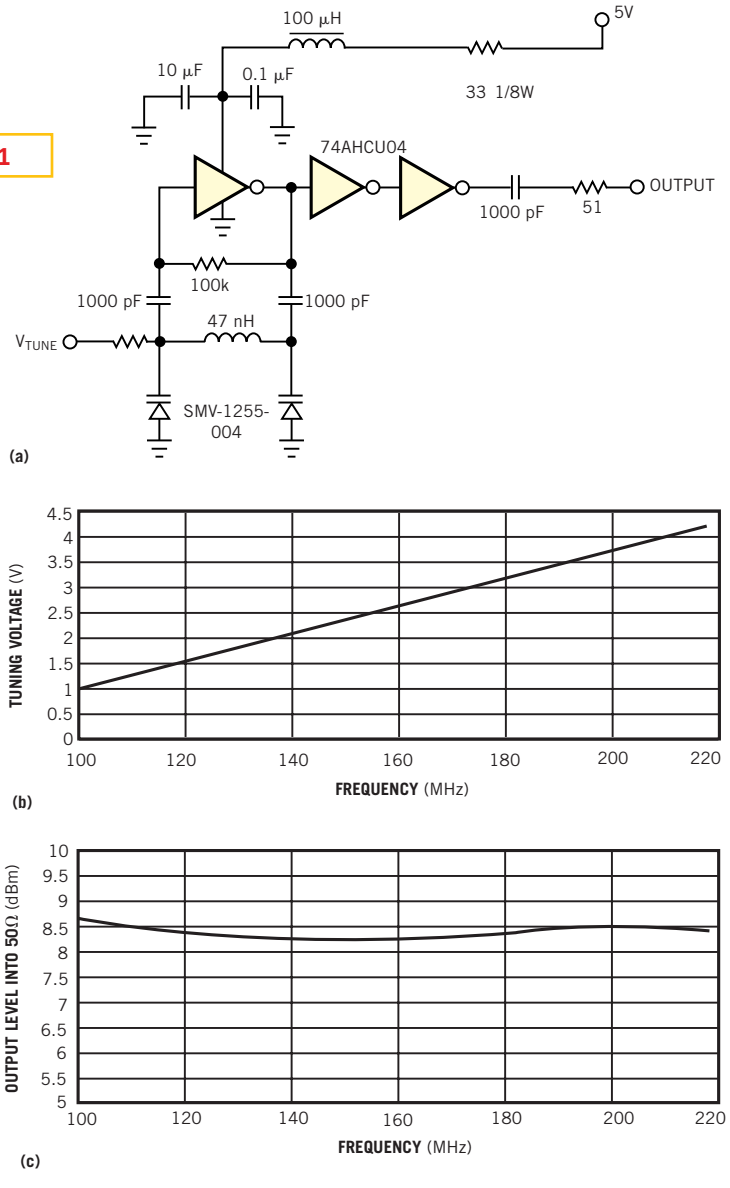
Shawn Stafford, AM Communications Inc, Quakertown, PA

A robust and VERSATILE VCO provides a stable output to 300 MHz (**Figure 1**). The circuit's simplicity, unconditional stability, and consistent high-drive capability over an octave make the oscillator ideal for many applications, such as synthesized sources, local oscillators, and transmitters. The AHC logic family (Texas Instruments, www.ti.com) makes the circuit's performance possible. AHC is a relatively new line of CMOS logic whose high speeds and good noise performance allow oscillator operation into regions in which bipolar-junction-transistor and FET designs prevail.

Figure 1

The oscillator topology is a modified Colpitts oscillator for which two hyperabrupt varactor diodes create the capacitive divider. The SMV-1255-004 (Alpha Industries, www.alphaind.com) encloses two varactors in one SOT-23 package (**Figure 1a**). The capacitance-voltage ratio of these varactors allows linear tuning over an octave with less than 4V (**Figure 1b**). You can substitute other varactors as long as the loaded Q of the resonant circuit is high enough to ensure start-up oscillation, but tuning characteristics may change. The inductor is a wound spring type chosen to maximize resonant Q. Oscillation is unstable when you use a low-Q, surface-mount-wound, chip-type inductor. The 100-kΩ resistor biases the gate to provide the gain and the 180° of phase shift necessary for oscillation. A lowpass filter with a low-frequency cutoff is highly recommended on the IC's power pin. Without this filter, incidental modulation from power-supply noise and pickup easily contaminate the oscillator signal. A dedicated voltage regulator is also recommended in noisy environments, but the filter is still necessary to keep the signal as clean as possible.

With a 5V supply, current consumption is approximately 25 mA±1 or 2 mA, depending on the frequency of oscillation. Using a 33Ω series resistor can reduce the current to 18 mA and supply enough power for reliable oscillation. The cascaded gates provide extra buffering and drive; the output resistor improves match with additional buffering. If your design needs a known constant output impedance, you can substitute a resistive match pad for the output resistor and maintain a considerable output level. **Figure 1c** shows the drive capability over frequency at mid-VHF, as well as level variation of less than 0.5 dB over the selected octave. Temperature effects on level are minimal with less than 1-dB change over 0 to 75°C, and worst-case harmonics are always better than –12 dBc. (DI #2294)



A 300-MHz VCO (a) uses varactor diodes with a capacitance-voltage ratio that allows linear tuning over an octave with less than 4V (b). A high-drive capability over an octave (c) makes the oscillator ideal for many applications.

To Vote For This Design, Circle No. 411

Pushbutton or logic controls nonvolatile DAC

Stephen Woodward, University of North Carolina, Chapel Hill, NC

FOR MANUAL CONTROL of analog signals, it's hard to beat the venerable precision multiturn potentiometer's simplicity, resolution, and power-off nonvolatility. When digital control of an analog parameter is the design objective, a universe of DACs is available to the designer. The circuit in **Figure 1**, however, has manual-pushbutton and CMOS/TTL-compatible digital interfaces to a 10-bit, nonvolatile, two- or four-quadrant multiplying DAC. The heart of the circuit is the Xicor (Milpitas, CA) X9511 PushPot series of digitally controlled potentiometers. These devices implement a convenient up/down response to either ground-referenced contact closures (with built-in debounce and pullup provisions) or open-collector/drain digital pulses.

Other useful features of these digital potentiometers include a ±5V analog-signal range and automatic storage and retrieval of settings with power-on/off cycles via an on-chip EEPROM. The potentiometer's only shortcoming in this context that its resolution is inadequate for precision applications (only 32 distinct settings, equivalent to a mere five bits). To overcome this limitation, the circuit combines two PushPots with a summing op-amp buffer to achieve nearly 10-bit resolution. IC_1 provides a weighted sum of the wiper voltages of P_2 (coarse input) and P_1 (fine input) in the ratio of 25.5-to-1. This operation provides a composite resolution of 32·(25.5+1)=848 distinct settings, equivalent to 9.7 bits.

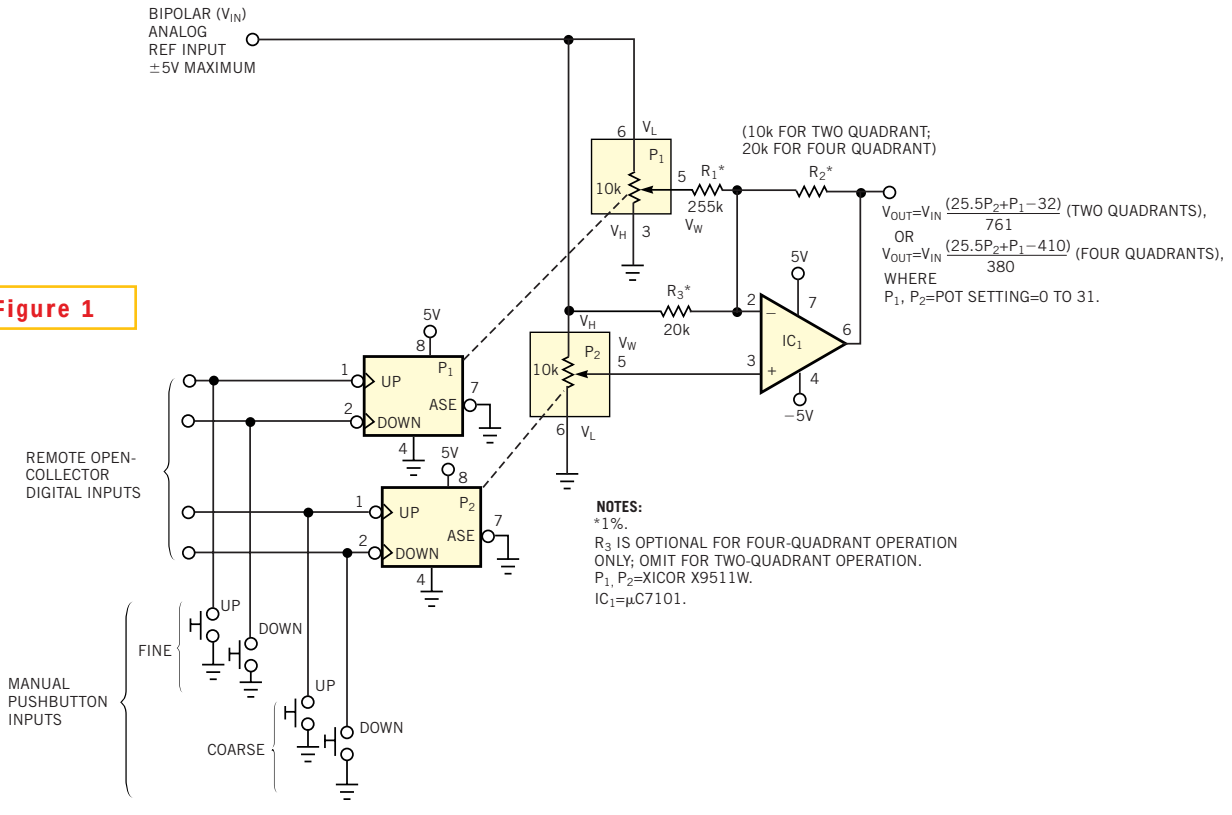
The missing 0.3 bits are lost to the good-but-still-only-finite differential linearity of the X9511 (Xicor specifies ±0.2 LSBs) and the consequent need to give a less-than-ideal weight (32×0.8 instead of 32) to P_2 to guarantee overall DAC monotonicity. The resultant two-quadrant (R_2=10 kV, R_3 omitted) gain equation is $V_{OUT}/V_{IN}=(25.5\times P_2+P_1-31)/761$. Thus, two-quadrant gain runs from –0.04 to 1.04 in steps of 0.0013, as P_1 and P_2 settings vary from (0,0) to (31,31).

Optionally, you can obtain four-quadrant multiplication by adding one resistor to the circuit, with the value $R_3=R_2=20$ kV. Gain then becomes $V_{OUT}/V_{IN}=(25.5\times P_2+P_1-410)/380$ and ranges from –1.08 to 1.08 in steps of 0.0026, as P_1 and P_2 vary from 0 to 31. The loading of P_1 by R_1 is light enough to produce a negligible effect on linearity. Connecting Pin 7 (automatic store enable) of P_1 and Pin 7 of P_2 to ground enables automatic storage of potentiometer settings to internal EEPROM upon power-down. The circuit then automatically retrieves the settings on power-up. (DI #2269).

To Vote For This Design,
Circle No. 412

Use a pushbutton or provide a digital signal to choose a nonvolatile analog output with nearly 10-bit resolution.

Figure 1



Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____

E-mail _____ **Fax** _____

Company _____

_____

Address _____

_____

Country _____ **ZIP**_____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____

Your vote determines this issue's winner. Vote now, by circling the appropriate number on the reader inquiry card.

design ideas

Edited by Bill Travis and Anne Watson Swager

Self-modifying code extends addressing mode

Paul Sofianos, Motorola Inc, Tempe, AZ

As just about any assembly-language programmer knows, self-modifying code (SMC) is usually undesirable, unintentional, and destructive. However, the SMC routine in **Listing 1** is extremely useful to extend the indexed, 16-bit offset addressing mode of the venerable HC05 µC from 8 bits, or 256 locations, to the full 13-bit (8-kbyte) memory address space that the µC's architecture supports. This routine is very useful for error tables, text messages tables, or any array manipulation for a large number of elements. (You can download **Listing 1** from EDN's Web site: www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2280.)

Indexed addressing with offset is useful for selecting an element of byte length, L, in an N-element table. In general, you calculate the effective addresses of this element as follows:

EA=TA+L3N+EO,

where EA=effective address of the memory location; TA=the absolute address of the start of the element table; L=number of bytes associated with each table element; N=the desired element number, beginning with zero; and EO=element offset, which includes all integer values from 0 to L–1.

The HC05 calculates the quantity L3N+EO and places the result in the index register. The µC then adds this dynamic value to the static quantity, TA, to arrive at the effective address. Unfortunately, the index register of the

LISTING 1—SELF-MODIFYING-CODE SUBROUTINE

```
EXAMP   EQU     $03             ;Example table element (0 to N) to transfer from the
SIZE    EQU     $0E             ;For this example, all elements have length 14 (base 10).
                                ;TABLE to TARGET

*******************************************************************************
*               Ram Table                                                     *
*******************************************************************************
        ORG     $0050
SMCRAM  RMB     $4              ;Self-modifying-code subroutine
TEMPX   RMB     $1              ;Temp IDX storage
TEMPA   RMB     $1              ;Temp ACC storage
TARGET  RMB     SIZE            ;Target location of element length SIZE

*******************************************************************************
*               Initialization                                                *
*******************************************************************************
        ORG     $0400           ;Start of rom
INIT    LDA     SMCROM          ;Initialize the self-modifying-code.
        STA     SMCRAM          ;LDA instruction
        LDA     SMCROM+$3
        STA     SMCRAM+$3       ;RTS instruction

*******************************************************************************
*               Program start                                                 *
*******************************************************************************
BEGIN   LDX     #EXAMP          ;Transfer the example element
        JSR     XFER            ;Transfer all bytes from the TABLE rom to TARGET ram
        STOP

*******************************************************************************
*               Subroutines                                                   *
*******************************************************************************
*               Subroutine XFER
*               This routine transfers an element of the array TABLE(X) to TARGET.
*               The length of each element, in bytes, is SIZE (1<=SIZE<=255).
*               The number of the element to be transferred is in IDX.
*               Both ACC and IDX are returned intact.
XFER    STX     TEMPX           ;Temporarily save IDX
        STA     TEMPA           ;Temporarily save ACC
        LDA     #SIZE           ;Calculate the effective address of the first
                                ;byte for the desired element.
        MUL
        ADD     SMCROM+$02      ;Add the element offset to the TABLE offset
        STA     SMCRAM+$02
        TXA
        ADC     SMCROM+$01
        STA     SMCRAM+$01
        CLRX                    ;Begin transferring the individual bytes, starting with the
                                ;byte located at the effective address and ending with the byte
                                ;located at the effective address + SIZE -1
GETBYTE JSR     SMCRAM          ;Get the byte by extended addressing
        STA     TARGET,X        ;Save in the TARGET ram
        INC     SMCRAM+$02      ;Add 1 (double-precision) to the SMCRAM extended address
        BNE     NOINC
        INC     SMCRAM+$01
NOINC   INCX                    ;Update the index counter
        CPX     #SIZE           ;Quit when SIZE bytes have been transferred
        BNE     GETBYTE
        LDX     TEMPX           ;Restore IDX
        LDA     TEMPA           ;Restore ACC
        RTS

*               Subroutine SMCROM (Dummy)
*               This dummy SMCROM (Self-modifying-code ROM) routine is transferred
*               to SMCRAM during program initialization. Extended address TABLE, the starting
*               address of the data table, is used to calculate an absolute address of a byte
*               of data in the table which is then transferred into ACC.
SMCROM  LDA     TABLE           ;Absolute addressing pointing to the start of the
        RTS                     ;message table

*******************************************************************************
*               Data table                                                    *
*******************************************************************************
*               This is the actual data table which can contain any number of entries (up to
*               the limits of the HC05's rom space), with all entries of length SIZE.
TABLE   FCB     'TABLE_ENTRY_00'
        FCB     'TABLE_ENTRY_01'
        FCB     'TABLE_ENTRY_02'
        FCB     'TABLE_ENTRY_03' ;Selected element for this example
*               .
*               .
*               .
        FCB     'TABLE ENTRY 99'

*******************************************************************************
*               Vectors                                                       *
*******************************************************************************
        ORG     $1FFE
RES     FDB     INIT            ;Reset vector
```

HC05 is only 8 bits long, limiting L3N+EO to values of 0 to 255, which is inadequate for large tables.

The SMC routine in **Listing 1** easily overcomes this limitation. Simply put, the routine copies a dummy static-command set that employs extended addressing from ROM to RAM. The code calculates an effective address for EO=0 and stores the result as an absolute address, such as TA, for this RAM command. The code then executes this RAM command, fetching a single byte from ROM at an absolute, extended address and saving the byte in RAM by using indexed, 8-bit offset addressing. The routine fetches successive memory locations from the data table and places the contents in the target table in RAM. This process continues until the transfer of all L bytes of the desired element is complete. (DI #2280)

To Vote For This Design,
Circle No. 371

VHDL procedure dynamically opens a file

Jacques Behar, Rockwell Semiconductor Systems, San Diego, CA

A simple VHDL-87 procedure opens a file whose name a command file or user conveys in runtime. You can apply this technique to the reading or writing of data, control, and status files. This procedure is useful for replacing the input stimuli file of a testbench without recompiling the code (with a different stimuli file name) or rename the stimuli file name. In addition, the procedure can write the simulation results to a file whose name consists of the input file name and any preferred extension.

The example in **Listing 1** shows the skeleton of a program that reads and sums a file of integers. In this example, the program first reads the input data file name from the command file "name.cmd" in the first four lines after "begin." Alternatively, the user can interactively enter the data file name. A VHDL procedure embedded in a concurrent VHDL process achieves the dynamic file access. The main process uses the READ_NEW_INT signal to activate the concurrent READ_FILE process in this example (**Listing 2**). Upon activation, the READ_FILE process invokes the READ_FILE_ PROCEDURE. The FILENAME signal conveys the name of the read file. Note that the length of the file name is also necessary, and the FILENAME_LEN signal conveys this length to the READ_FILE procedure.

The MAIN process has sole control over each access to the input file, which is obvious in this simple example that reads a new integer each time the MAIN process toggles the READ_NEW_INT signal. In some complex cases, the MAIN process could conditionally open a file after part of the simulation is complete. At the end of the file, the procedure exits automatically, and the file closes.

You can download both listings from EDN's Web site: www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2281. (DI #2281)

To Vote For This Design,
Circle No. 372

LISTING 1–VHDL-87 PROCEDURE

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use std.textio.all;

-----------------------------------------
entity READ_FILE is
end READ_FILE;

-----------------------------------------
architecture JB of READ_FILE is

signal  READ_FILE_ENDED : std_logic := '0';
signal  FILENAME        : string(1 to 80) := (others => ' ');
signal  FILENAME_LEN    : integer;
signal  SINT            : integer;
signal  READ_NEW_INT    : std_logic := '0';


begin

--------------------------
MAIN: process

    file     Fname   : text is in "name.cmd"; --script file
    variable VLINE   : line;
    variable VSTR    : string(1 to 80) := (others => ' ');
    variable VSUM    : integer := 0;

    begin
        readline(Fname, VLINE);
        FILENAME_LEN <= VLINE'length;
        read(VLINE, VSTR(1 to VLINE'length));
        FILENAME     <= VSTR;                  --filename to process

        while (READ_FILE_ENDED /= '1') loop
            READ_NEW_INT <= not READ_NEW_INT; --get new read
            wait for 10 ns;                   --set the read cycle time
            VSUM := VSUM + SINT;              --sum numbers read as an example
        end loop;
        wait;

end process MAIN;


end JB;

-----------------------
-- configuration
-----------------------
configuration CFG_READ_FILE of READ_FILE is
    for JB
    end for;
end CFG_READ_FILE;
```

LISTING 2–VHDL READ_FILE PROCESS

```
RD_FILE: process
    variable VLINE  : line;
    variable VINT   : integer;

    procedure READ_FILE_PROCEDURE(fname: string) is
        file     ifile : text is in fname;
    begin
        while (NOT endfile(ifile) ) loop
            readline(ifile, VLINE);
            read(VLINE,VINT);
            SINT <= VINT;
            if (endfile(ifile)) then exit; end if;
            wait on READ_NEW_INT;
        end loop;
    end READ_FILE_PROCEDURE;

    begin
        wait until READ_NEW_INT = '1';
        READ_FILE_PROCEDURE(FILENAME(1 to FILENAME_LEN));
        READ_FILE_ENDED <= '1';
end process RD_FILE;
```

Configure buck converter for boost operation

Mehrzad Koohian, Semtech Corp, Newbury Park, CA

Buck converters are inherently different from boost converters, because buck converters typically use the high side of the output as the power switch's reference. However, a buck converter with a floating output drive section is configurable as a boost controller (**Figure 1**). This circuit configures the SC1101 buck controller for a 5 to ±12V boost with ±500 mA of output current. The BST pin, which normally connects to a high-side drive supply in a buck converter, connects to V_{CC} to drive the ground-referenced MOSFET. By tying PGND to circuit ground, the SC1101 becomes a boost controller, yielding 12V from 5V. An output charge-pump voltage inverter provides –12V at 0.5A as well.

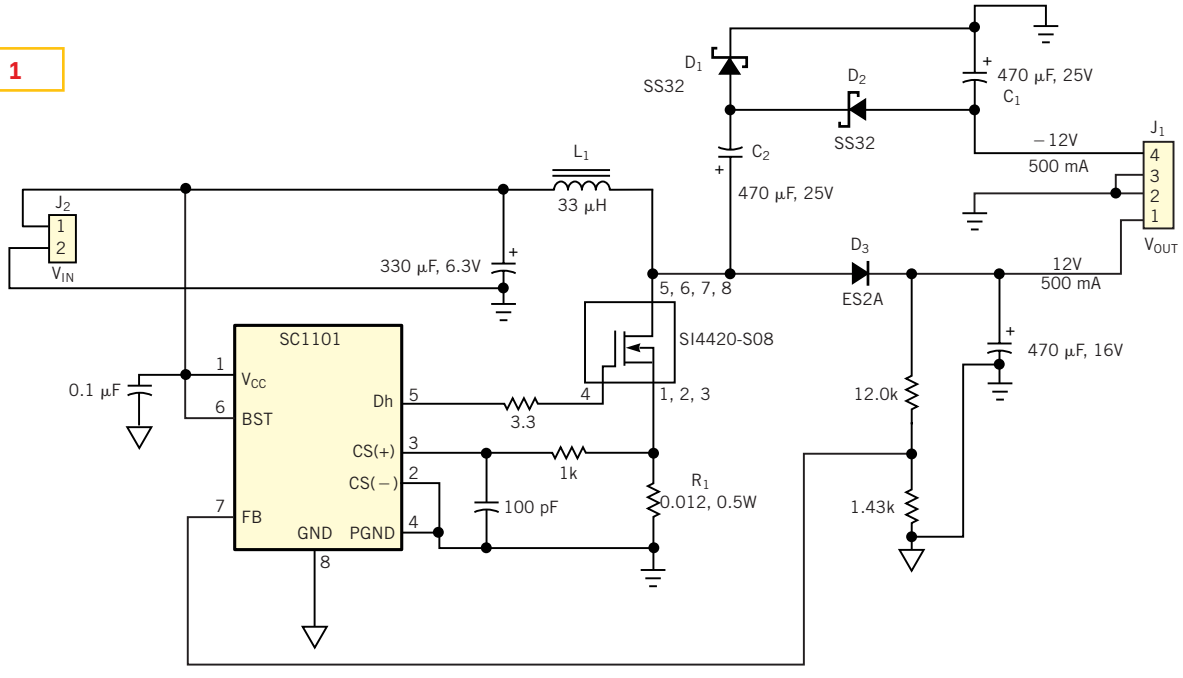
The sense resistor, R_1, serves two purposes. First, it assures proper start at power-up with full load by limiting the duty cycle. With the output capacitors not charged, a full-load condition demands high peak inductor currents. If the duty cycle exceeds a maximum limit, the inductor does not have a chance to discharge and will saturate. By limiting the peak currents and thus the duty cycle, the output capacitors can charge in several cycles upon start-up, thus preventing inductor saturation. R_1 also limits switch current during an overload or short circuit. In this application, a value of 0.012V provides for peak-current limiting and allows the circuit to deliver the required output current of ±500 mA.

A fast silicon rectifier, D_3, rather than a Schottky diode, rectifies the 12V output. Use of the silicon rectifier balances the voltage drops in the –12V rectifier circuit, which is two Schottky-diode drops, with the 12V rectifier drop. Because the application is power-limited, the circuit can trade off current that the –12V supply draws for current on the 12V output. If the –12V output is unused, the 12V output can deliver 1A, and you can eliminate D_1, D_2, C_1, and C_2. To improve efficiency under this condition, you can also use a Schottky diode for D_3. The controller provides separate grounds for power drive reference (PGND) and analog-circuitry common (GND). These two grounds must connect at the controller.

With dual outputs, the circuit's measured efficiency is 91% with V_{IN}=5V and 93% with V_{IN}=5.5V. In **Figure 1**'s circuit, L_1 consists of a #T37-52 core (Micrometals Inc, www.micrometals. com) with 34 turns of 26-gauge wire. For applications that exceed ambient temperatures of 50°C, you can use a slightly larger core, such as a #T38-52, or a Kool MU core material available from Magnetics Inc (www.mag-inc.com). (DI #2279)

To Vote For This Design,
Circle No. 373

Figure 1



Tying a buck converter's BST pin to V_{CC} and the PGND pin to circuit ground configures the converter for boost operation.

Circuit eases three-phase monitoring

Henno Normet, Tavares, FL

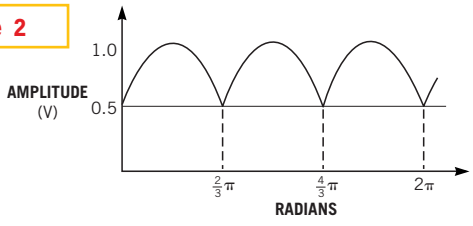
Measuring line-to-line voltages in a delta-connected three-phase system can present special problems. Because all three lines may be floating several hundred volts above ground, you can not use nonisolated, grounded oscilloscopes or other single-ended instruments. Special isolation amplifiers are available for oscilloscopes, but they can cost several thousand dollars. You still need to make three measurements even with proper instrumentation. The circuit in **Figure 1** reduces the magnitude of the line-to-line voltages and combines them into one ground-referenced signal (**Figure 2**) that you can safely monitor with a grounded oscilloscope.

Phase C serves as a floating common (ground) for the circuit. Unity-gain buffer amplifiers IC_1 and IC_2 prevent loading of the 30-to-1 voltage dividers connected from A to C and from B to C. Op-amp IC_3 is a differential-input instrumentation amplifier that provides a signal proportional to the voltage between A and B. Unity-gain amplifier IC_4 inverts the B-C signal such that the half-wave-rectified signals can combine with the proper 120° phasing. The forward voltage drops across D_1 to D_3 cause a small error. You can minimize this error by using germanium diodes (1N34s), which have lower voltage drops than their silicon counterparts, and by keeping the signal levels as high as possible.

All the circuitry floats at power-line potential; thus, it is dangerous to monitor the rectifier outputs. To obtain a safe output, you should use an ISO122P isolation amplifier. The ISO122P is a unity-gain amplifier with an output that is fully isolated from its input. You need two line-isolated ±15V supplies to power the ISO122P and the op amps. The circuit has additional applications, the details of which are beyond the scope of this Design Idea. For example, in some cases, it is important that the three phase voltages are equal (balanced); their absolute values may be of less importance. It is much easier to detect an imbalance looking at a single waveform than it is looking at three waveforms. For continuous unattended monitoring, the design needs only one undervoltage/overvoltage detector, rather than one detector for each of the three phases. (DI #2283).

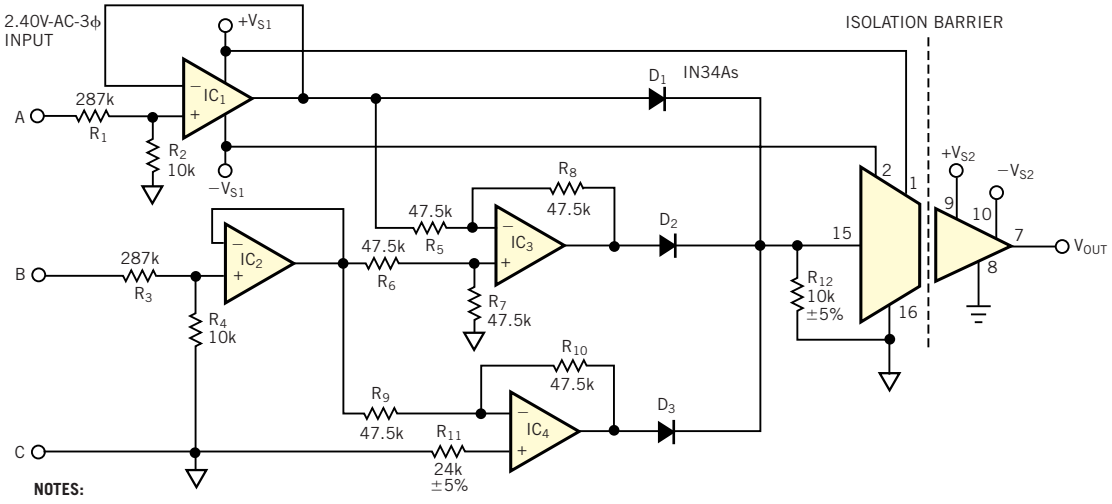
To Vote For This Design, Circle No. 374

Figure 2



The output of the isolation amplifier in Figure 1 is a low-level, ground-referenced voltage.

Figure 1



NOTES:

=FLOATING GROUND.

=SYSTEM GROUND. (DO NOT TIE THESE POINTS TOGETHER.)

ALL RESISTORS EXCEPT R_{11}, R_{12}=±1% METAL FILM.
ALL OP AMPS=1/4LM324.
ISOLATION AMPLIFIER=BURR-BROWN ISO122P.
CAUTION: LETHAL VOLTAGES ARE PRESENT IN THE CIRCUIT.
USE EXTREME CARE.

A quad op amp combines the three half-wave-rectified phase voltages into one voltage for easy monitoring.

PLL forms simple MSK demodulator

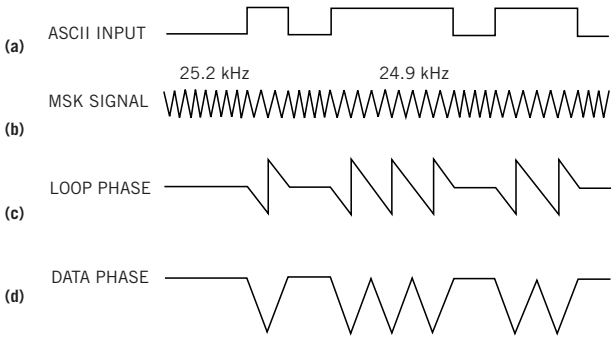
Tom Napier, North Wales, PA

In minimum-shift-keying (MSK) signaling, two frequencies that differ by the bit rate represent a one bit and a zero bit. Normally, the frequency shift occurs at the peak of a cycle, so that neither the amplitude nor the slope of the waveform shows a discontinuity. We needed to transmit 300-baud ASCII text using ultrasonic transducers. These devices have a very narrow bandwidth around their 25-kHz resonant frequency, making MSK the obvious choice for modulation. A zero bit becomes 84 cycles of 25.2 kHz, and a one bit is 83 cycles of 24.9 kHz. It is easy to generate this signal with a PIC µC and an 8-bit DAC. However, a traditional MSK demodulator circuit uses a center-frequency VCO and several mixers and filters. This design needs something simpler: to wit, the circuit in **Figure 1**.

Because the transmitter sends 25.2 kHz between characters, the design phase-locks a 74HCT4046 PLL chip to this frequency. The chip has three phase detectors, each with different characteristics. By choosing the correct two, you can demodulate the MSK input without losing phase lock on the zero-bit carrier. Phase Detector 3 on the PLL chip has a 360° linear range. That is, its mean output varies from 0 to 5V and then switches back to 0V as the phase passes through 360°. Adjust the frequency of the PLL chip so that it locks to 25.2 kHz with a 180° phase error and an output of 2.5V. If the oscillator frequency remains fixed, then you can recognize a one bit by its phase error, which swings from 0 to 360° during the bit.

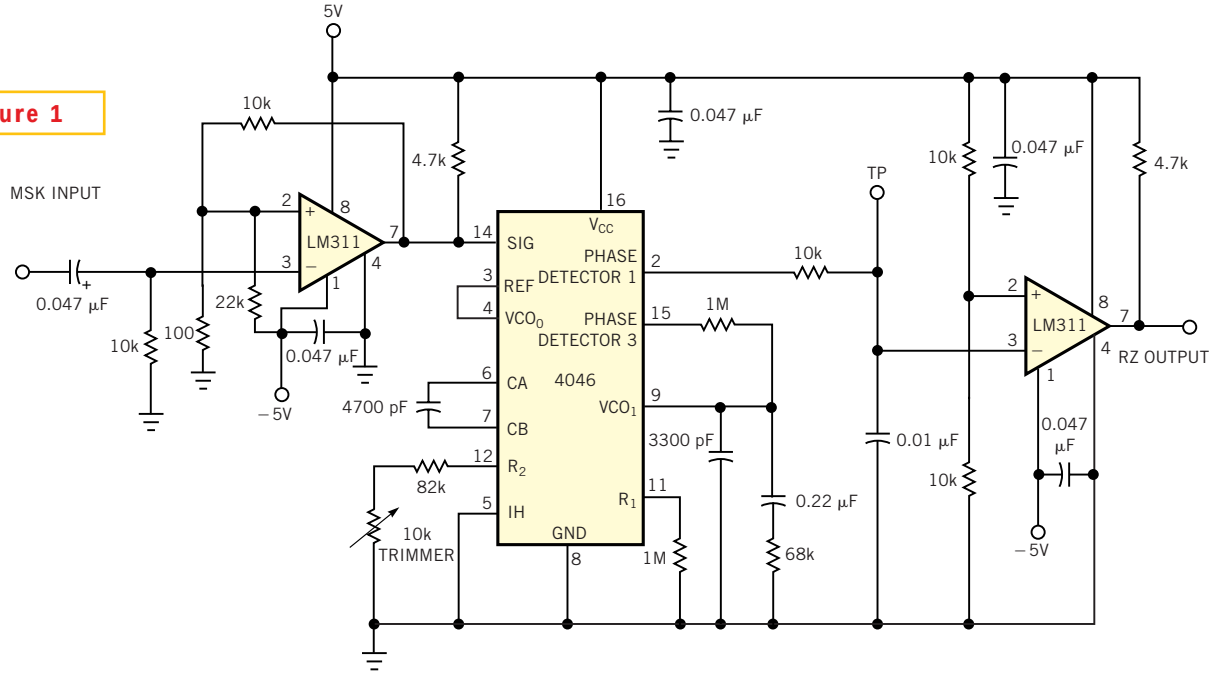
Because the initial lock is at the 180° point, a one bit results in a phase error that goes from 180° down to 0°. It then jumps

Figure 2



The ASCII input character for the letter "g" (a) produces a frequency-varying MSK signal (b). The phase error as seen by the loop (c) produces the phase-error signal at TP (d) that drives the output.

Figure 1



A PLL makes MSK demodulation inexpensive and easy.

to 360° and continues back down to 180°. The net result is a ramp that goes down to 0V, a jump to 5V, and then another downgoing ramp. The mean voltage is 2.5V. Because the loop bandwidth is approximately 15 Hz, the instantaneous effect on the VCO is small, and the net frequency change is zero. Rather than detect the ramp-jump-ramp waveform, use Phase Detector 1 as the data output. Because this block is a simple exclusive-OR gate, each one bit appears as a spike going from 5 to 0V and back. A comparator can change it into a return-to-zero version of the input signal. Alternatively, the output can go to a retriggerable monostable with a 3.5-msec period to generate a good approximation of a nonreturn-to-zero output. **Figure 2** shows the circuit waveforms that occur for the letter "g." (DI #2284).

To Vote For This Design,
Circle No. 375

µC makes inexpensive sine-wave generator

Jorge Luis B Romeu, IdeaWorks LTD, Syracuse, NY

You can use A/D converters or external, controllable oscillators to generate sine waves from low-power, low-cost µCs. However, these methods add cost, reduce reliability, increase circuit and software complexity, increase power consumption, and increase overall size. Alternatively, and with just a few lines of code, most µCs can easily generate multiple discrete sine waves. The example in **Figure 1** uses a 68HC705J1A to generate sine waves of 9 to 20 kHz. The circuit uses the µP's square-wave output and switches between multiple RC filters of varying cutoff frequencies to achieve outputs with reasonable spectral purity.

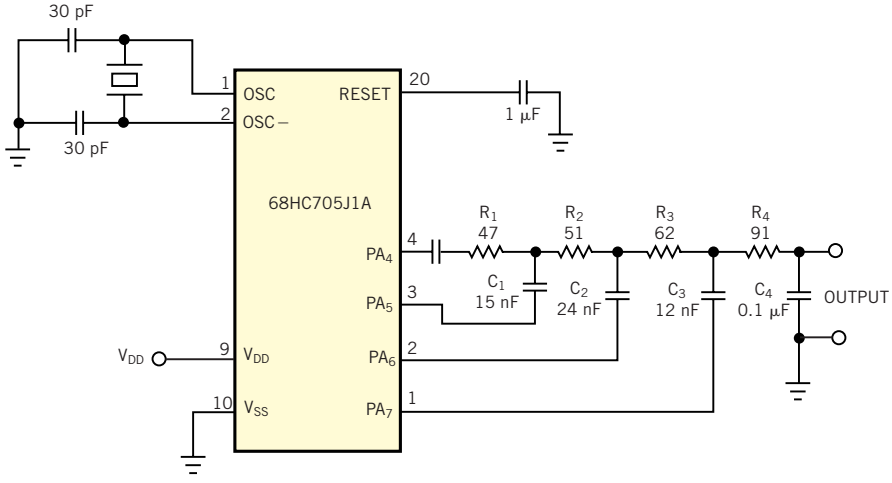
The necessary code consists of a simple subroutine that can adapt to different needs, such as tone duration and multiple (sequential) tone output (**Listing 1**). Moreover, the generated frequency is based on a variable, "frec," that a previous routine can pass to this subroutine. The duration time is based on a timer, which eliminates calculations of tone duration based on the cycle period. The code in **Listing 1** is for illustrative purposes, but you can use it as-is with proper headers. (You can download **Listing 1** and an example calling subroutine from *EDN*'s Web site: www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2278.)

The basic circuit in **Figure 1** uses only four output pins of the µC; three pins are for filter returns, and the fourth is the µP output. The filter constants are such that the highest cutoff frequency corresponds to the highest generated frequency. The

TABLE 1–EXAMPLE FREQUENCIES AND CORRESPONDING NUMBER OF LOOPS

Filter number and cutoff frequency (kHz)	Number for variable, "Frec" (=number of loops)	Actual output frequency (kHz)	Filter number active (R#, C#)
1: 10	14	9.5	1,2,3,4
2: 12	11	11.75	2,3,4
3: 15	8	15.35	3,4
4: 18	7	17.1	4 (always on)

Figure 1



Based on a simple subroutine, a mC can easily generate multiple sine waves from 9 to 20 kHz by switching external RC filters on and off.

subsequent cascaded filters cut off at lower frequencies, and the circuit can switch these filters in or out depending on the desired output (**Table 1**). The µC's speed, code efficiency, the number of discrete frequencies to be generated, and the spacing between those frequencies all determine the maximum frequency of operation. The number of loops for one-half cycle of output frequency (50% duty cycle) equals the number of cycles per loop×time/cycle. With a 3.58-MHz crystal, the time per software cycle is 558 nsec. (The 705J1A µC uses one-half of the oscillation frequency for its internal operating frequency.) **Table 1** shows some example frequencies and the corresponding number of loops.

The RC filters have a cutoff frequency, or –3-dB point, at the generated frequency for maximum linearity and minimum distortion. You should note that the output is approximately the same level across all frequencies because the resistors are always in series when driving a high-impedance load. However, because of leakage when a capacitor is active, or grounded, the lowest frequency has a lower output than the highest frequency. You should optimize the R and C values in **Figure 1** for your application, desired output frequencies, and impedance. You can switch the filters on and off sequentially, independently, or in combinations to suit different needs. (DI #2278)

LISTING 1–FREQUENCY-GENERATING SUBROUTINE

```
fgen    sta temp1               ;store accumulator in a temp variable
        lda #04                 ;number of interrupts for 250mS (5*65mS)
timer   bset 2,tscr             ;clear real time interrupt flag
innr3   bset 4,output           ;turn on oscillator output
        ldx frec                ;number for one half cycle
innr1   decx                    ;count down for half cycle
        bne innr1               ;if not finished, keep counting
        bclr 4,output           ;turn off oscillator output
        ldx frec                ;number for other half of cycle
innr2   decx                    ;count down for half cycle
        bne innr2               ;if not finished, keep counting
        brclr 6,tscr,innr3      ;if 65mS not passed (see timer/counter
section of docs) do another cycle
        deca                    ;subtract from the five  65mS for 250mS
total
        bne timer               ;have 250mS gone by? If not go back and
repeat all
        lda temp1               ;restore accumulator
        rts                     ;
```

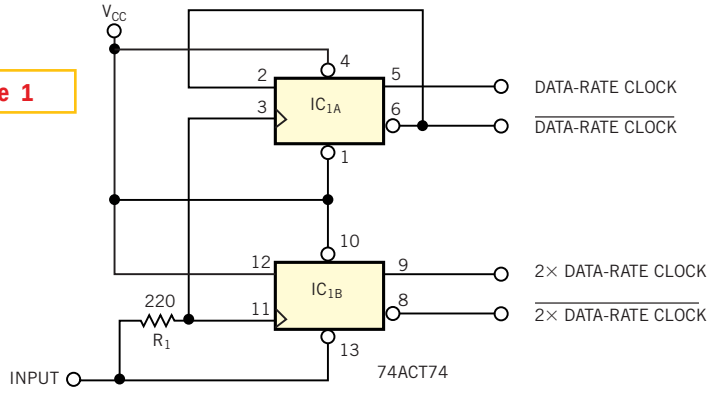
To Vote For This Design,
Circle No. 376

74ACT74 makes low-skew clock divider

Tom Napier, Consultant, North Wales, PA

Serial-data systems often generate an internal clock at twice the data rate for mid-bit sampling or for generating bi-phase codes. External equipment and some internal processes require a clock that runs at the data rate. Simply dividing the twice-rate clock with a flip-flop generates a data-rate clock that is skewed by one logic delay with respect to the input. This delay can be a significant fraction of the bit period. You can use specialized PLL-based low-skew divider chips to deal with this problem, but these chips have a limited frequency range and are not designed to follow rapid changes in the data rate.

Figure 1



The 74ACT74 dual flip-flop generates two clock rates with negligible skew.

The circuit in **Figure 1** uses a dual flip-flop, the 74ACT74, to generate both clock rates as well as both clock polarities with negligible skew. One half of the chip, IC_{1A}, acts as a normal divide-by-two circuit. The other half, IC_{1B}, tracks the input clock because the input's leading edge triggers IC_{1B} high and the input's trailing edge resets IC_{1B}. The divider transitions are synchronous within a few hundred picoseconds with the positive transitions of the twice-rate clock that the chip's other half generates. This circuit works with inputs from a few hertz to more than 100 MHz.

Without R_1, the input removes the reset from the twice-rate flip-flop at the same moment as the input clocks this flip-flop on. Theoretically, this setup is allowable because the 74ACT74's reset-recovery time is specified as 0 nsec. In practice, a resistor in the 100 to 500V region, in conjunction with the chip's input capacitance, delays the clock inputs slightly and adds a useful safety margin. (DI #2282)

To Vote For This Design,
Circle No. 377

design ideas

Design Idea Entry Blank

Entry blank must accompany all entries. $100 Cash Award for all published Design Ideas. An additional $100 Cash Award for the winning design of each issue, determined by vote of readers. Additional $1500 Cash Award for annual Grand Prize Design, selected among biweekly winners by vote of editors.

To: Design Ideas Editor, EDN Magazine
275 Washington St, Newton, MA 02158

I hereby submit my Design Ideas entry.

Name _____

Title _____

Phone _____ **Fax** _____

E-mail _____

My e-mail address may be published Yes ____ **No** ____

Company _____

_____

Address _____

_____

Country _____ **ZIP** _____

Design Idea Title _____

_____

Social Security Number _____
(US authors only)

Entry blank must accompany all entries. (A separate entry blank for each author must accompany every entry.) Design entered must be submitted exclusively to EDN, must not be patented, and must have no patent pending. Design must be original with author(s), must not have been previously published (limited-distribution house organs excepted), and must have been constructed and tested. Fully annotate all circuit diagrams. Please submit software listings and all other computer-readable documentation on a IBM PC disk in plain ASCII.

Exclusive publishing rights remain with Cahners Publishing Co unless entry is returned to author, or editor gives written permission for publication elsewhere.

In submitting my entry, I agree to abide by the rules of the Design Ideas Program.

Signed _____

Date _____